MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AD-A151 254

AFOSR-TR- 85-0179

③

Grant No. AFOSR-81-0170

Annual Final Report

"Feasibility Studies of Optical Processing
of Image Bandwidth Compression Schemes"

by

B. R. Hunt

R. H. Strickland

July 15, 1984

DTIC
SELECTED
MAR 1 3 1985
E

DTIC FILE COPY

ENGINEERING EXPERIMENT STATION
COLLEGE OF ENGINEERING
THE UNIVERSITY OF ARIZONA
TUCSON, ARIZONA

AD-A151254

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY Unclassified	**3. DISTRIBUTION/AVAILABILITY OF REPORT**
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
	AFOSR-TR- 85-0179

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
The University of Arizona		AFOSR/NE

6c. ADDRESS (City, State and ZIP Code)	7b. ADDRESS (City, State and ZIP Code)
Engineering Experiment Station College of Engn Tucson, AZ 85721	Bldg 410 Bolling AFB DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR	NE	AFOSR-81-0170

8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.			
Bldg 410 Bolling AFB DC 20332-6448	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
	61102F	2305	B1	

11. TITLE (Include Security Classification) FEASIBILITY STUDIES OF OPTICAL PROCESSING OF IMAGE BANDWIDTH COMPRESSION SCHEMES

12. PERSONAL AUTHOR(S)
B. R. Hunt, R. H. Strickland

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Annual	FROM Mar83 TO Mar84	July 15 1984	57

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

See Reverse

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
ROBERT W CARTER, Lt Col, USAF	(202) 767-4932	NE

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

The research currently sponsored under Grant No. AFOSR-81-0170 consists of several distinct and separate activities. The separate research efforts are unified by a common theme: the application of optical processing for image bandwidth compresssion. Within this common theme, however, the separate research projects are not completely related to each other. Therefore, this report is put together, literally, as a number of independent reports. The separate sections of the report, which follow this section, are intended to be read separately and independently of any other section. Each section has its own references and its own figure labellings, for example.

The separate sections of this report, and the research problems dealt with in each section, are summarized in the following:

(1) Data compression by multi-spectral staggered sampling, and data reconstruction by spatial and spectral interpolation; (see Section III of report).

(2) Data compression by optical tomography, with data reconstruction by optical convolution and back projection; (see Section IV of report).

(3) Adaptive data compression by spatial transformations to create a spatially stationary image; (see Section V of report).

(4) Improvement of the optical data compression method known as IDPCM, (see Section VI of report).

keywords: Computations, algorithms.

UNCLASSIFIED

Grant No. AFOSR-81-0170

Annual ~~Final~~ Report

"Feasibility Studies of Optical Processing

of Image Bandwidth Compression Schemes"

by

B. R. Hunt

R. H. Strickland

July 15, 1984

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

85 02 26 018

(I.) Introduction

Grant No. AFOSR-81-0170 has an objective which is well-summarized by the Grant title: "Feasibility studies of optical processing for image bandwidth compression schemes." It is the intent of research sponsored under this Grant to direct investigation into the following issues:

(a) formulation of alternative architechtural concepts for image bandwidth compression, i.e., the formulation of components and schematic diagrams which differ from conventional digital bandwidth compression schemes by being implemented by various optical computation methods;

(b) simulation of optical processing concepts for image bandwidth compression, so as to gain insight into typical performance parameters and elements of system performance sensitivity;

(c) maturation of optical processing for image bandwidth compression until the cverall state of optical methods in image compression becomes equal to that of digital image compression.

It is the last of these, item (c), which represents the continuing strategic objective of the efforts being carried on under Grant No. AFOSR-81-0170. It is important to remember that the major attention given to image bandwidth compression has been for methods most conveniently implemented by digital computations. As flexible and multipurpose are digital methods, there

1

may always be operational circumstances, environments, or con-
straints where the availability of a different technology is
important. However, with the concentration upon digital compu-
tations, which has characterized most research on bandwidth
compression, alternative methods in optics have suffered. Thus,
the purpose of research sponsored under this Grant is to serve
as a source of alternatives for future concepts in bandwidth
compression, so that the environment for compression technology
need not be dominated by one methodology.

2

(II.) Overview of the Report

The research currently sponsored under Grant No. AFOSR-81-0170 consists of several distinct and separate activities. The separate research efforts are unified by a common theme: the application of optical processing for image bandwidth compression. Within this common theme, however, the separate research projects are not completely related to each other. Therefore, this report is put together, literally, as a number of independent reports. The separate sections of the report, which follow this section, are intended to be read separately and independently of any other section. Each section has its own references and its own figure labellings, for example.

The separate sections of this report, and the research problems dealt with in each section, are summarized in the following"

(1) Data compression by optical tomography, with data reconstruction by optical convolution and back projection (see Section III).

(2) Adaptive data compression by modification of a previously demonstrated technique, IDPCM, to an efficient spatially recursive scheme (see Section IV).

(3) Adaptive image processing by using spatial transformations to create a spatially stationary image (see Section V).

(III.) Tomography and the Projection Matrix

Tomography is a procedure which decomposes a two-dimensional image into a series of one-dimensional projections, each made at a different angle through the original image. A projection is produced by integrating the image data in one direction across the image. Along the horizontal axis, for example a projection is defined by:

$$P_\theta(x) = \int_{-\infty}^{\infty} f(x,y)dy$$

where P_θ is the projection at angle θ and $f(x,y)$ is the original image. Subsequently, either the coordinate system or the object is rotated and the next projection is calculated.

In digital tomography, the summed (integrated) data results are registered by a one-dimensional array of discrete sensors or detectors. This string of detectors must be large enough to record all of the data at each of the possible angles. For a square image, the maximum number of detectors is required at the angles of 45° and 135°; see Figure 1. At other angles however, the ends of the detector array swing outside of the image, thus registering artificial zero data. The scalloped ends of the projection matrix, visible in Figure 2, are the result of these artificial zeros.

The projection matrix of any image displays certain consistent characteristics. As seen in the projection matrix of Figure 2, there is a sinusoidal design woven into the matrix; this pattern is present in the projection matrix derived from any image.

Each horizontal line in the projection matrix corresponds to the projection data gathered by all N detectors at one specific angle; a horizontal line, therefore, is referred to as a data sequence from "within"
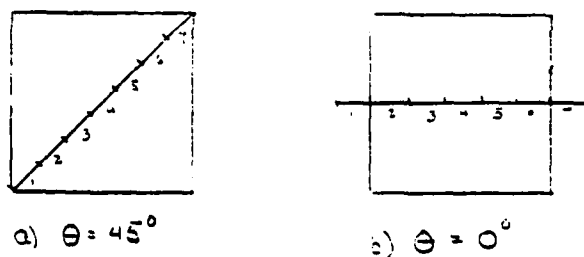
a) $\theta = 45°$ b) $\theta = 0°$

Figure 1: demonstrates how the number of detectors necessary varies as a function of projection angle. At a projection angle of 45° (a) seven detectors are required, but at a projection angle of 0° (b), only five detectors are needed; the extra detectors at the endpoints swing out of the image.
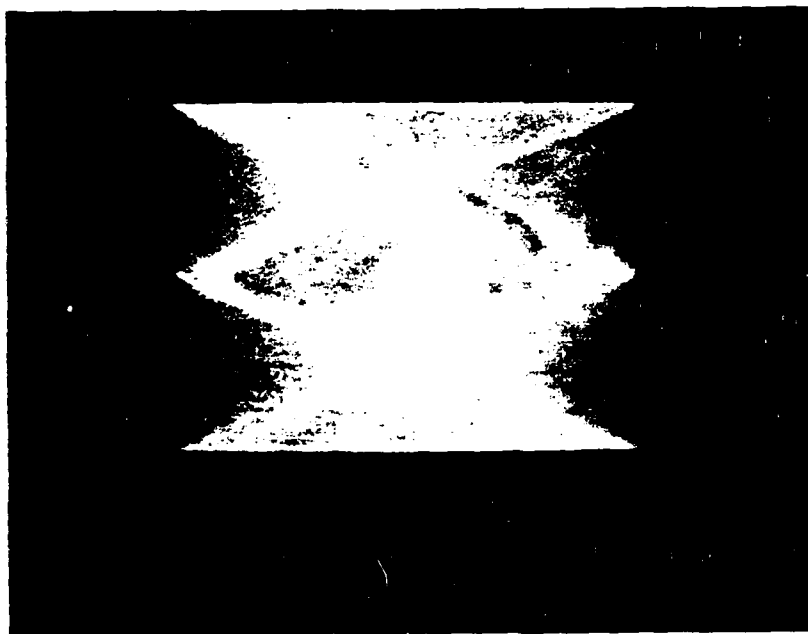

Figure 2: Projection matrix or sinogram

a projection (See Figure 3). Conversely, a vertical line through the projection matrix represents the data integrated at <u>one</u> detector as it travels through all M angles; this is referred to as data "across" projections (See Figure 4).

To reconstruct the original image, filtered back-projection is done. Each projection first is convolved with a high pass, rho filter. Following this, the filtered projections are spread back across the image plane at their original angle and then the results from all angles are summed.

Redundancy in a Projection Matrix

The purpose of data compression techniques is to remove, or at least reduce, the redundant or correlated information, thereby retaining only orthogonal or decorrelated data. It is well known that the data in most images if highly redundant. From this knowledge, one might surmise that the projection matrix also contains a great deal of redundant information. To test this, a method was devised to measure the entropy in a projection matrix, both within and across projections. The information or entropy, H, can be calculated using:

$$H_\theta = \sum_{i=1}^{n} p_{\theta_i} \log_2 \frac{1}{p_{\theta_i}} \quad \text{bits} \quad \text{(within projections)}$$

$$H_d = \sum_{i=1}^{n} p_{d_i} \log_2 \frac{1}{p_{d_i}} \quad \text{bits} \quad \text{(across projections)}$$

where n is the number of cells in the data histogram, θ is a specific angle, d is a specific detector and p_{θ_i} or p_{d_i} is the probability of data being contained in the ith cell of the histogram.
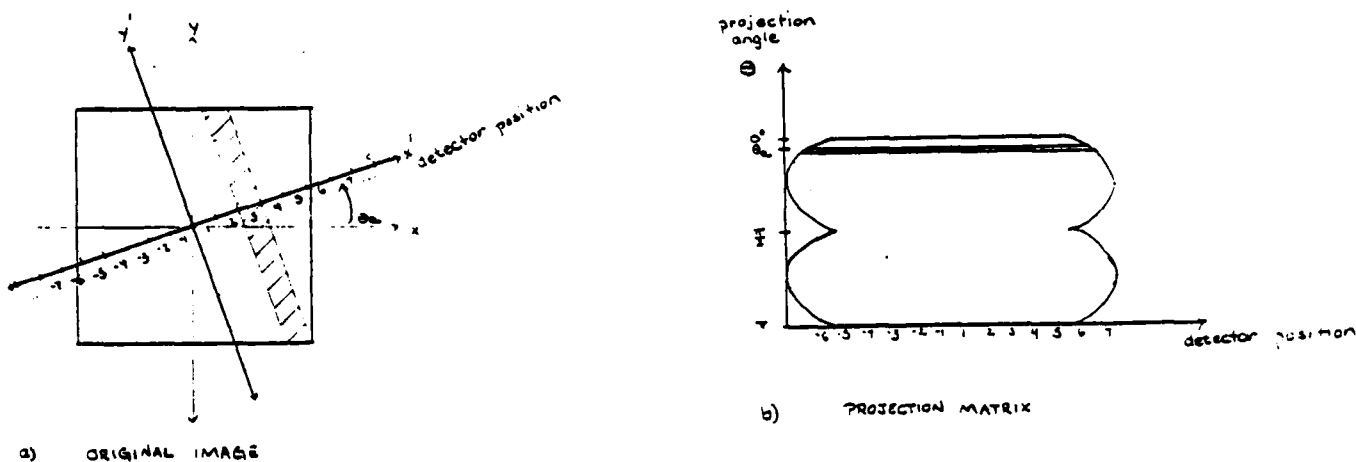
a) ORIGINAL IMAGE

b) PROJECTION MATRIX

Figure 3: The values of the original image (a), integrated along y' are sensed by each of the detectors along x'; for example, the result at detector number three is the sum of values in the shaded region of (a). For the projection angle Θ_a, the results from each detector are placed into the the projection matrix (b) along a horizontal line.
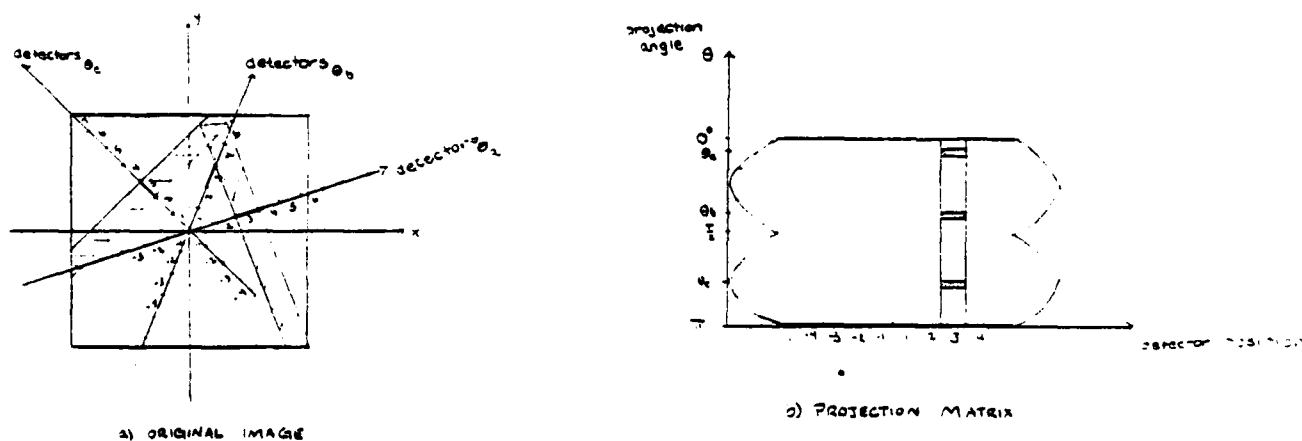

a) ORIGINAL IMAGE

b) PROJECTION MATRIX

Figure 4: A vertical line in the projection matrix (b),e.g., along detector number three, represents, in the original image (a), the integrated result from a single detector (in this example, number three) as it swings through all possible angles.

The redundancy in a line of projection data then can be calculated:

$$R_\Theta = H_{max} - H_\Theta \quad \text{(within)} \quad \text{or} \quad R_d = H_{max} - H_d \quad \text{(across)}$$

where H_{max} is the maximum amount of information possible.

Method

The original image, the source of the projection data, was 128x128 pixels in size. For every projection angle, 182 detectors (each detector being 1 pixel in width) were used and a total of 100 projections were made. For ease in calculation, the data in the matrix, with an original range from 0 to 34323, were scaled to a range from 0 to 255. A histogram of data values was computed for each line in the matrix. The histogram was divided into 16 cells, each with a width of 16 intensity levels. The maximum amount of information possible would be present if all cells in the histogram were equally probable:

$$H_{max} = \log_2 16 = 4 \text{ bits}$$

Results

Figure 5 shows the redundancy <u>within</u> each projection as a function of projection angle. The amount of redundancy appears to be strongly dependent upon the angle of projection. The redundancy is highest near the angles of 0°, 90° and 180° and is lowest near angles of 45° and 135°. The maximum obtained redundancy of 2.184 (54.6% of the maximum) occurs at an angle of 0° and the smallest value of 0.0577 (1.4% of maximum) is at the angle of 135°.
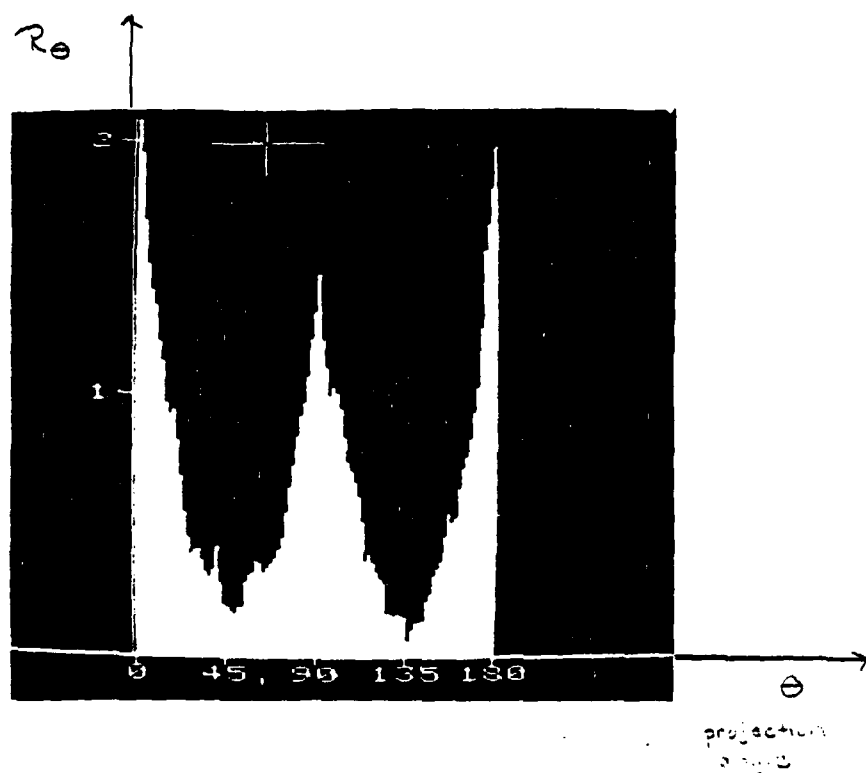
Figure 5: Redundancy within projections, as a function of projection angle.

Greater redundancy at certain angles primarily appears to be a function of how the tomographic process operates on a square image. Figure 6 shows why such an effect may occur.

Redundancy <u>across</u> projections is shown in Figure 7. Detectors near the endpoints of the projection matrix, those that swing out of the image near angles of 0°,90° and 180° were excluded from analysis to avoid the introduction of artificial zero data. The values vary, for the most part between redundancies of 1 to 2 bits (25-50%) for each detector.

Compression of Projections using DPCM

Differential pulse code modulation (DPCM) coders operate on the principle of quantizing prediction error values rather than actual data values. Based on the recent history of the signal, a predictor $\hat{x}(n)$ is made to approximate the actual $x(n)$. Then, rather than quantizing and storing $x(n)$, the difference:

$$d(n)=x(n)-\hat{x}(n)$$

is quantized and stored. Subsequently, the decoder attempts to recover the original signal by essentially integrating the quantized differences between samples. Figure 8 shows a block diagram of the basic closed loop DPCM system.

For a data sequence $x(n)$ with a normalized autocovariance, $\rho \rightarrow 1$, the variance of the difference signal, which is the input to the quantizer, is significantly smaller than that of the original data. Since the quantization error variance is directly proportional to the variance of the quantizer input, it is possible to lower the quantization bit rate to a specific
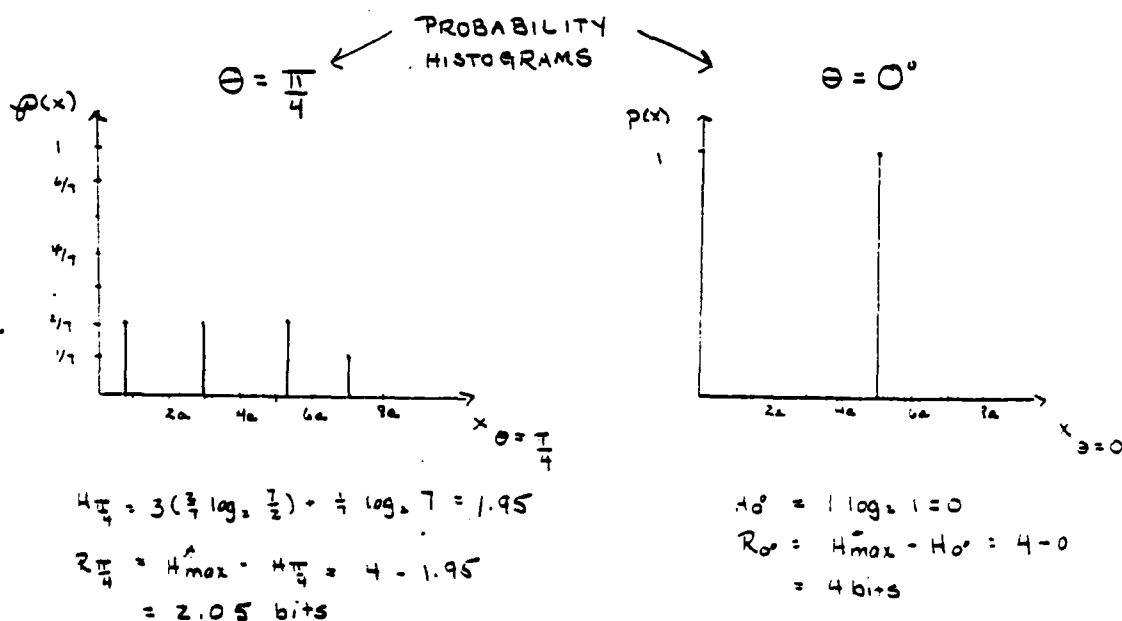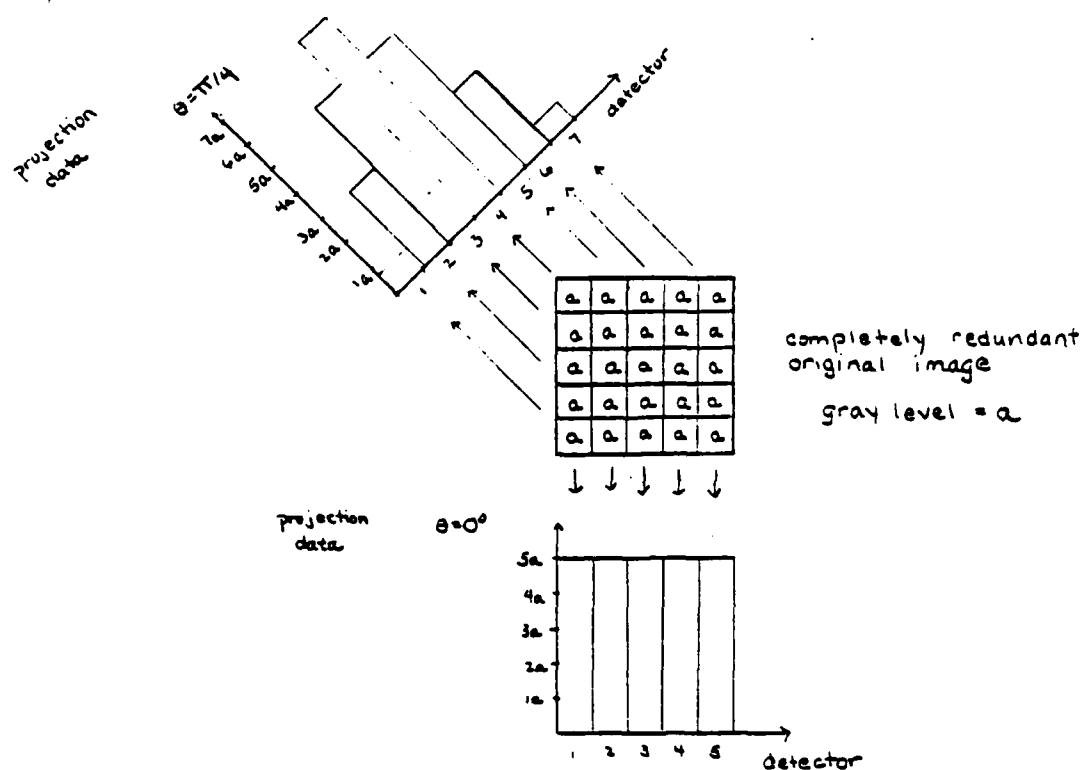
Figure 5 demonstrates how redundancy varies as a function of projection angle due to the tomographic process operating on a square image. Assuming a completely redundant original image, the projection data at $\theta = 0°$ is totally redundant also; however, the projection at $\theta = 45°$ is only about 50% redundant.
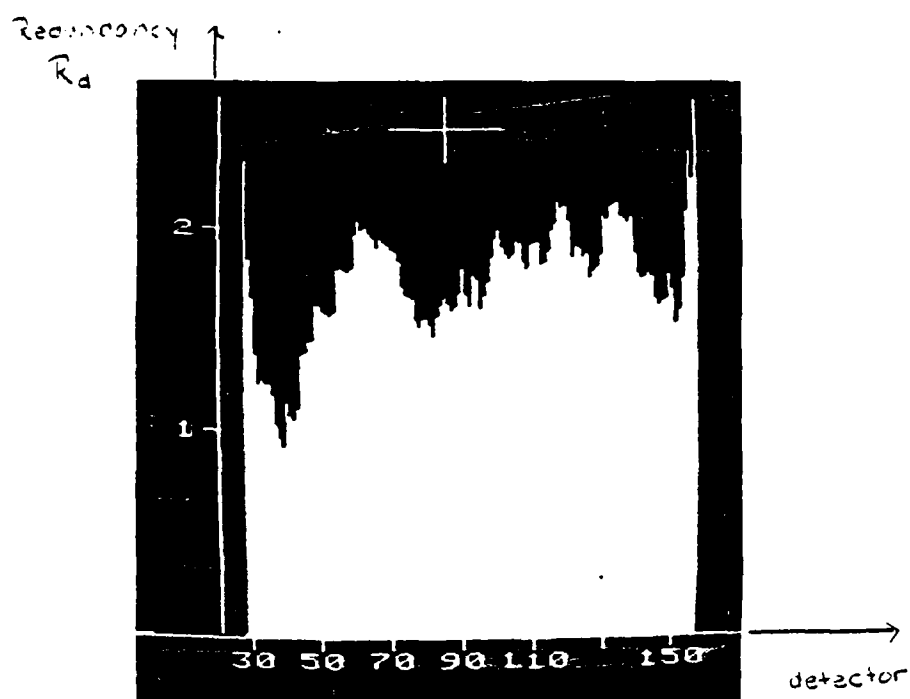
Figure 7: Redundancy across projections as a function of detector position.
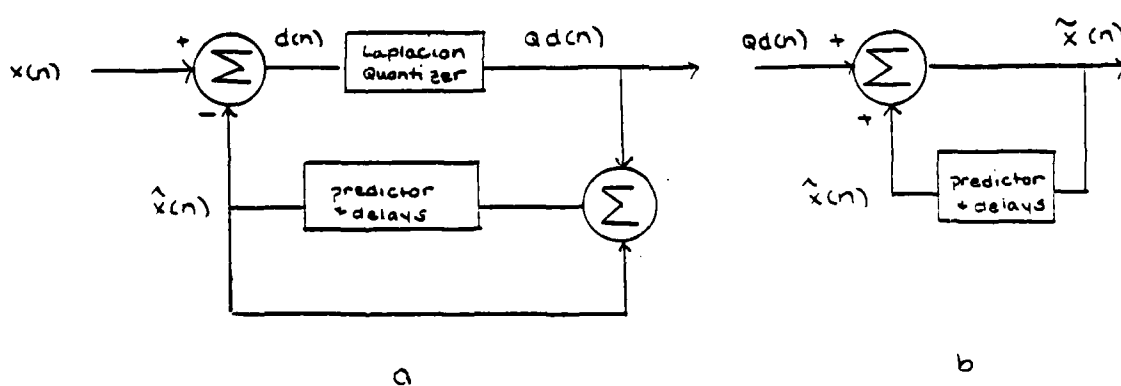

a

b

Figure 8: Block diagram of DPCM coding (a) and decoding (b) system.

level and maintain the same signal to noise ratio (SNR). Further reductions
in the bit rate, past the new level, also are possible but at the expense
of the SNR.

The closed loop or feedback-around quantizer diagram of Figure 8
ensures that the quantization errors do not accumulate into subsequent
samples, i.e., the errors should be independent.

There are a variety of prediction schemes which can be used. The
two used in this study are:

1. Normalized autocovariance or rho method

2. Average rate of change or slope method.

Prediction Methods

For both of the prediction methods, coding of the projection matrix is
carried out one line at a time. The first value in each line is carried
through the system at its full bit rate and the remaining values in the line
are quantized at the reduced rate.

The prediction errors, the input to the quantizer in a DPCM system,
tend to be distributed as a Laplacian (exponential) probability density
(Gray, 1983). Decision and reconstruction levels have been derived to
minimize the mean square error in quantizing a Laplacian. These levels,
given in Pratt (1978), are used in the quantizer of Figure 8. The mean
square error between the original and decoded values of the projection
matrix were computed for each line and subsequently were averaged across
lines.

In order to perform DPCM on lines of constant sample size, the scalloped
ends of the projection matrix were trimmed before encoding, producing a

rectangular matrix. The data values which were removed were those collected by the detectors farthest from the origin, those that swing out of the image at angles near 0°, 90° and 180°. Tomographic reconstruction without these data produces a circular image.

Rho Prediction. The Rho method involves a linear predictor of the form:

$$\hat{x}(n) = \sum_{j=1}^{n} h_j x(n-j)$$

and the quantizer input for a first order predictor is:

$$d(n) = x(n) - h_1 x(n-1)$$

Taking the square and expected value of the equation above, it can be shown:

$$\sigma_d^2 = \sigma_x^2 (1 - 2h_1\rho_1 + h_1^2)$$

where: ρ_1 is normalized autocovariance

To minimize the mean square error, take the partial derivative of the variance with respect to h_1 and set it equal to zero.

$$\frac{\partial \sigma_d^2}{\partial h_1} = \sigma_x^2 (-2\rho_1 + 2h_1) = 0$$

$$h_1 = \rho_1 \qquad \text{for minimum MSE}$$

Estimation of x by this first order predictor assumes that you have no other information concerning the expected shape of the projection data. The slope method, however, relies on the fact that some general assumptions can be made concerning the projection matrix.

Slope Method. As mentioned earlier, all projection matrices share the sinogram pattern. Based on this, it should be possible to make a more accurate estimate of x based not only on recent signal history but also on where x lies in the matrix and on whether DPCM is being carried out within or across projections.

The trimmed projection matrix is partitioned into a set of blocks, see Figure 9. For each block, an estimate is made of the average rate of change between $x(n)$ and $x(n-1)$, either across or within projections. This slope value is then used as a predictor in both the coding and decoding stages:

$$\hat{x}(n) = x(n-1) + slope(m,i)$$

where $x(n-1)$ is contained in the block (m,i) of the matrix. After their calculation, these slope values are retained for the decoding phase, consequently adding to the final bit rate for the coded matrix.

Results

Table 1 gives the mean square error for each prediction method and for various bit rates. Before compression, the trimmed projection matrix contained values ranging from a minimum of 6776 to a maximum of 34323 with a mean of 21670. Therefore, a mean square error (MSE) of 100,000., for example, represents an average error of 317 per element of the matrix. For the minimum value this represents an average of $\pm$ 4.7% error, while for the maximum value, it is an average error of $\pm$ 0.92%.

Table 1

a) Within Projections

Rho Prediction		Slope Prediction	
MSE	bit rate	MSE	bit rate
54,745	4.09	41,681	4.20
115,784	3.10	74,831	3.21
270,901	2.11	158,945	2.22
986,390	1.12	483,260	1.23

Figure 9:   Trimmed projection matrix which has been partitioned into a set of blocks.   Each block, fcr example, may be 10x10, meaning 10 angles long and 10 detectors wide.

Table 1 cont.

b) Across Projections

Rho Prediction		Slope Prediction	
MSE	bit rate	MSE	bit rate
20,220	4.13	16,176	4.23
80,437	3.14	54,698	3.25
307,745	2.14	228,544	2.25
2,540,996	1.15	1,233,708	1.26

Once the compressed projections have been decoded, the next step is
to reconstruct the original image by back-projection. The following
photographs show the image in its original and reconstructed forms after
being subject to various amounts of data compression and different
prediction schemes.

Figure 10

a) original image

b) reconstructed image,
no data compression,
16 bits/element in matrix

c) coding within projections
4.09 bits/element in matrix
rho prediction method

d) coding within projections
4.2 bits/element in matrix
slope prediction method

Figure 10 cont.   19

e) coding within projections
   3.1 bits/element in matrix
   rho prediction method

f) coding within projections
   3.21 bits/element in matrix
   slope prediction method

g) coding within projections
   2.22 bits/element in matrix
   slope prediction method

Figure 10 cont.

h) coding across projections
   4.13 bits/element in matrix
   rho prediction method

i) coding across projections
   4.23 bits/element in matrix
   slope prediction method

j) coding across projections
   3.14 bits/element in matrix
   rho prediction method

k) coding across projections
   3.25 bits/element in matrix
   slope prediction method

A post hoc analysis of the error distribution, the input to the quantizer, was done to determine whether the assumption of a Laplacian distribution was justified. Figure 11 shows the error [d(n)] distributions resulting from both rho and slope prediction methods. The rho distribution contains obvious deviations from an exponential, but the slope distribution, by visual analysis, appears to match an exponential quite well. The main difference between the slope distribution and a theoretical Laplacian $[p(x) = \frac{a}{2} \exp(-a|x|)]$ occurs at the peak of the distribution and in the tails. The mode of the obtained distribution is lower than expected, while the tails are somewhat higher. Analysis of the obtained slope distribution using Pearson's Chi Square Goodness of Fit Statistic, however, indicates that it cannot be accurately modeled as a Laplacian ($p > 0.995$). Based on this finding, the question arises as to whether a better way to quantize the errors is possible. Preliminary results obtained in answering that question suggest that better quantization can be achieved when the decision and reconstruction levels are extended further towards the tails of the distribution.

Figure 12 shows the results of extending the quantizer levels outward. The quantization for this image was performed manually, by visual analysis of the histogram. This reconstruction, with a bit rate of 2.1 bits per element of the projection matrix, appears to be superior to the image of Figure 10g which is a slightly higher bit rate (2.22 bits); in fact Figure 12 seems to be almost comparable to the quality of some images quantized to approximately 3 bits. Although a sharper image is obtained by adjusting the quantization levels, a large amount of noise still appears in the reconstructed image.
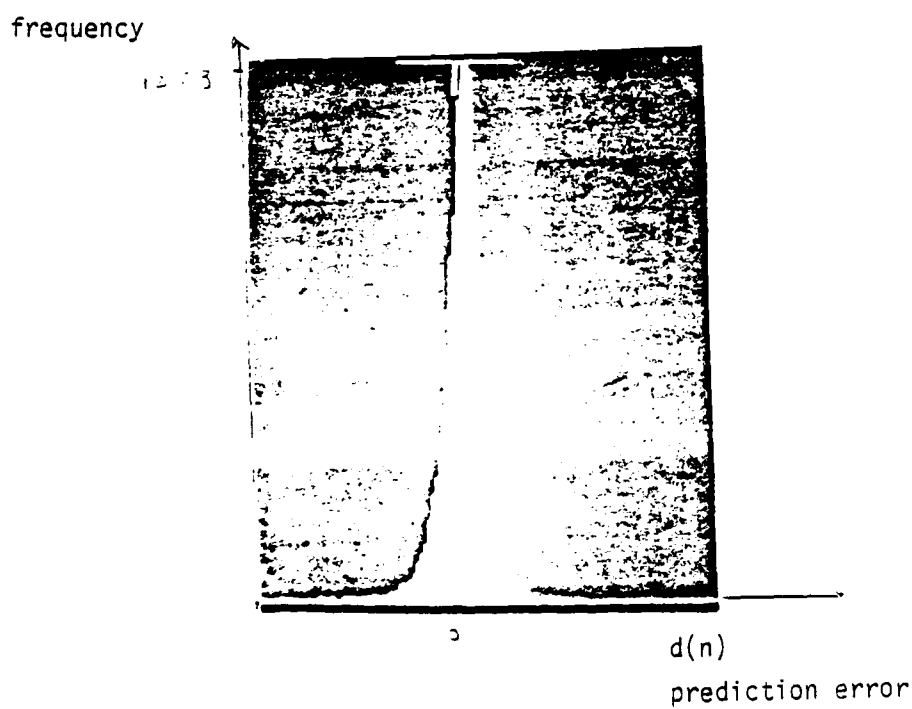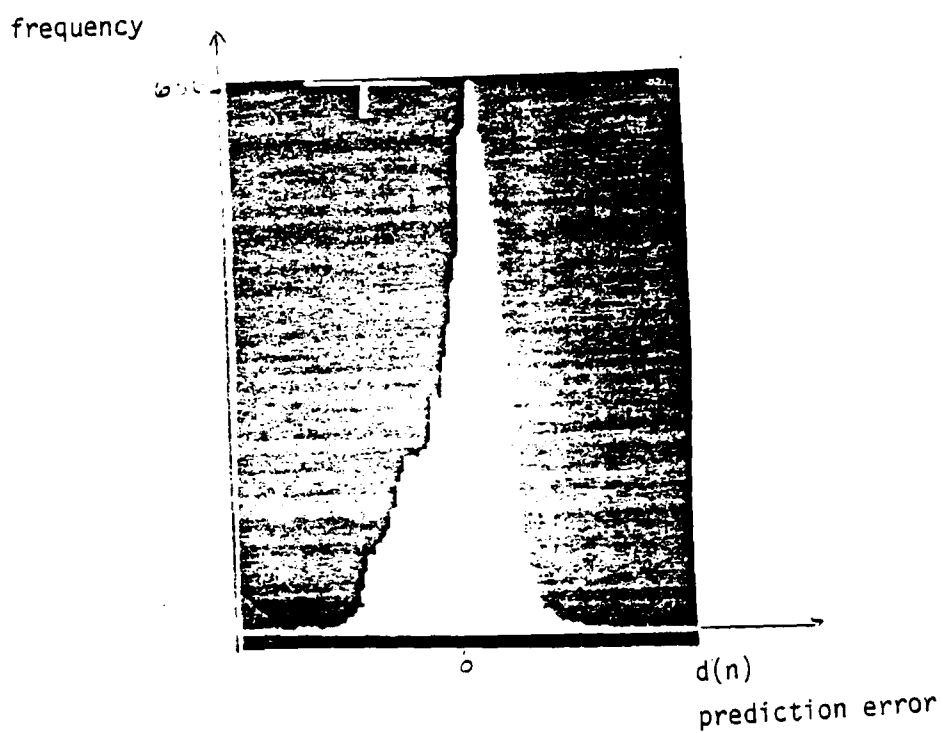
frequency

0 d(n)
prediction error

frequency

0 d(n)
prediction error

Figure 11: Distribution of prediction errors, d(n), resulting from the
two prediction methods; a)rho prediction, b) slope prediction.

Figure 12: Reconstruction from projections quantized by moving decision
and reconstruction levels out toward the tails of the error distribution.
Coding done within projections, bit rate of 2.1 bits/element in matrix
and slope prediction method was used.

Conclusion

Tomographic projections, originally 16 bit data, can be compressed to a bit rate of approximately 3.2 bits per element of projection matrix (a compression ratio of 5:1).and still produce a recognizable image; however, the image at this rate is of poor quality and unsuitable for most applications.

An improvement in results can be achieved by adjusting the decision and reconstruction levels of the quantizer. Another possible avenue for improvement would be to employ more sophisticated prediction procedures. However, a fundamental problem involved in this technique seems to rest upon the fact that any errors in the decoded projection matrix are amplified by the filtering done before back-projection. This high pass filter is, most likely, the cause of the noisy appearance of the recon-structed images. Since this filter is an integral part of back-projecting tomographic projections, it would appear that, while further research into this area may improve the results demonstrated here, the amount of improvement possible may not be significant enough to warrant the additional investigation.

References

Gray, Robert. <u>Multispectral Data Compression using Staggered Detector Arrays</u>.
  Doctoral dissertation, University of Arizona, 1983.

Pratt, William. <u>Digital Image Processing</u>. New York: Wiley & Sons, 1978.

IV.) Background Discussion

Image data compression methods can be classified in two basically different categories. One category is processing in the spatial domain. Another is "transform coding". In the first category are those methods which exploit redundancy in the spatial data. Redundancy is a characteristic which is related to predictability, randomness in the data. For example, an image of constant gray levels is fully predictable once the gray level of the first pixel is known. On the other hand, a white-noise random signal, such as that seen on a TV screen when no program is being broadcast, is totally unpredictable and every pixel has to be stored to produce the image.

Interpolated difference pulse-code modulation, developed by Hunt in 1977 [2], is a successful method of data compression in spatial domain. Since then, several variations of this method have been developed. Recently, a significant improvement to IDPCM, named recursive IDPCM, was demonstrated by Hunt and Cao [1].

Recursive IDPCM has two main features:

(1) This is a very efficient image data compression method which has achieved the result of bit rate below 0.4, and mean-square error below 0.002.

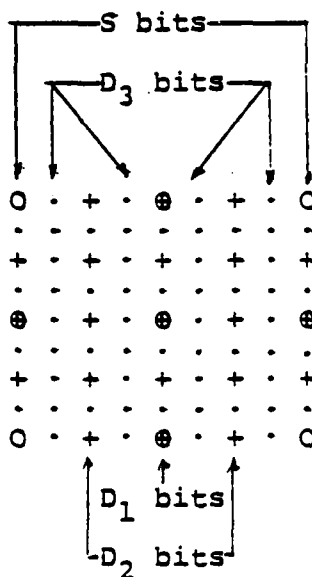(2) This method is quite simple and is very economical for machine cost [1].

A brief introduction to recursive IDPCM is shown below:

(1) Take coarse subsample spacing: N (e.g., N=8); quantize subsamples.

(2) Calculate the interpolated value of intermediate point between two subsamples; use this interpolated value to calculate the difference, and quantize the difference.

(3) Use the intermediate interpolated value added to the quantized difference to calcualte a pair of intermediate interpolated values with its neighbor two subsamples, etc.

A mode is defined as: S, DI, D2, D3, where S is the number of bits for quantizing subsamples, and DI, D2, and D3 are numbers of bits for quantizing differences. But DI, D2, and D3 have different numbers of bits. For example, take coarse subsample spacing N=8, as in the following:

```
          ┌──────S bits──────┐
          │ ┌──D₃ bits──┐    │
          │ │  ╲    ╱ │ ╲    │
          ▼ ▼   ▼  ▼   ▼  ▼  ▼
          O ·  +  ·  ⊕  ·  +  ·  O
          ·  ·  ·  ·  ·  ·  ·  ·  ·
          +  ·  +  ·  +  ·  +  ·  +
          ·  ·  ·  ·  ·  ·  ·  ·  ·
          ⊕  ·  +  ·  ⊕  ·  +  ·  ⊕
          ·  ·  ·  ·  ·  ·  ·  ·  ·
          +  ·  +  ·  +  ·  +  ·  +
          ·  ·  ·  ·  ·  ·  ·  ·  ·
          O  ·  +  ·  ⊕  ·  +  ·  O
                 │    ▲
                 │ D₁ bits │
                 └─D₂ bits─┘
```

8x8 subimage
O — subsamples using S bits quantization
⊕ — the first set quantized differences using D_1 bits
+ — the second set quantized differences using D_2 bits
· — the third set quantized differences using D_3 bits

Figure 1.

As explained above, this method is used to compress an entire image without regard to the amount of detail in any particular area. Generally, finer sampling is required in the neighborhoods of sharp gray-level transitions, while for relatively smooth regions, coarse sampling is acceptable. Although recursive IDPCM has a higher data compression ratio and low mean-square error than other methods, there is still room for improvement.

One improvement to recursive IDPCM is to use an adaptive scheme. With this method, an image can be divided into subimages, where a high bit rate is needed to deal with relatively complex subimages but a lower bit rate is sufficient for relatively simple subimages. Here, the complex subimages are defined as neighborhoods of sharp gray-level transitions and simple sub-images as neighborhoods of smooth gray-level transitions.

Identification of Subimages

In order to use an adaptive scheme, we have to detect the complexity level of each subimage. "One way to measure the redundancy of an image and to compare it to the nominal N^2p (NXN is the size of an image, p is bits of per pixel) bits is the use of the histogram statistics and the associated entropy statistics"[3]. Entropy represents the amount of information associated with the set of coder input values and gives a lower bound on the average number of bits required to code those inputs. If the set of coder input levels is W1, W2, W3,...Wm with probabilities p1, p2, p3,...pm, then it is not possible to code them without distortion using less than

$$S = -\sum_{i=1}^{m} p_k \log_2 p_k \text{ bits.}$$

However, this requires a very large amount of computation. Some approximation has to be introduced.
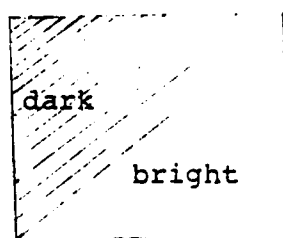
Measuring the level of sharp gray-level transitions can be done by calculating the function:

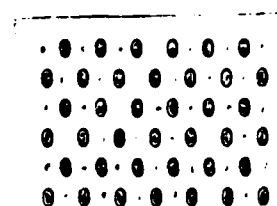$$R = \sqrt{\frac{\sum_{j=1}^{N} \sum_{i=1}^{N} (\overline{p} - p_{ij})^2}{N \times N}}$$

where $\overline{p}$ is the mean intensity of an NxN size subimage, and pij is the intensity of each pixel in a subimage. But for subimages a, b, and c shown in fig. 2, we get the same value for R.

a   b   c
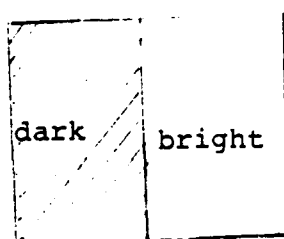
. =bright, ● = dark

Fig. 2

In fact, subimages a and b are in simple categories (low frequency) and the subimage c is in complex category (high frequency). The more simple an image is, the smaller the differences are between the image and a value which interpolates it. Therefore, complexity could be associated with the differences. The function,

$$R' = \sqrt{\frac{\sum_{i}^{m} (P_i - \hat{P}_i)^2}{M}}$$

can be used to approximately measure the complexity of a subimage, where, p_i is an original pixel and $\hat{p}_i$ is an interpolator of the subimage. M is the number of differences in a subimage. In recursive IDPCM, the middle interpolated image between two subsamples is subtracted from the original pixel and the first set of differences are quantized at N_1 bits. Then the middle interpolator added to its quantized difference and the neighboring subsample S are used to calculate the second set of interpolated values and their differences. Therefore, several quantization levels can be used both for the first set differences and second set differences. Fig. 10, 11, and 12 show that, differences, as defined above, can represent the complexity of subimages. This is an effective and feasible method, which is easy to compute.

Quantization

"A quantizer is a device whose output can have only a limited number of possible values. Each input is forced to one of the allowable output values. One way to accomplish this is to divide
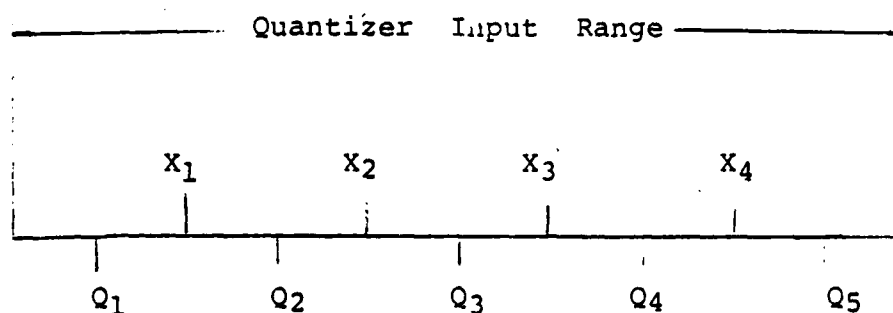
Quantizer Input Range

X_1 X_2 X_3 X_4

Q_1 Q_2 Q_3 Q_4 Q_5

Figure 5

In Recursive IDPCM, the typical distribution of differences is shown in Figure 6,

$P(x)$

$P(x)$
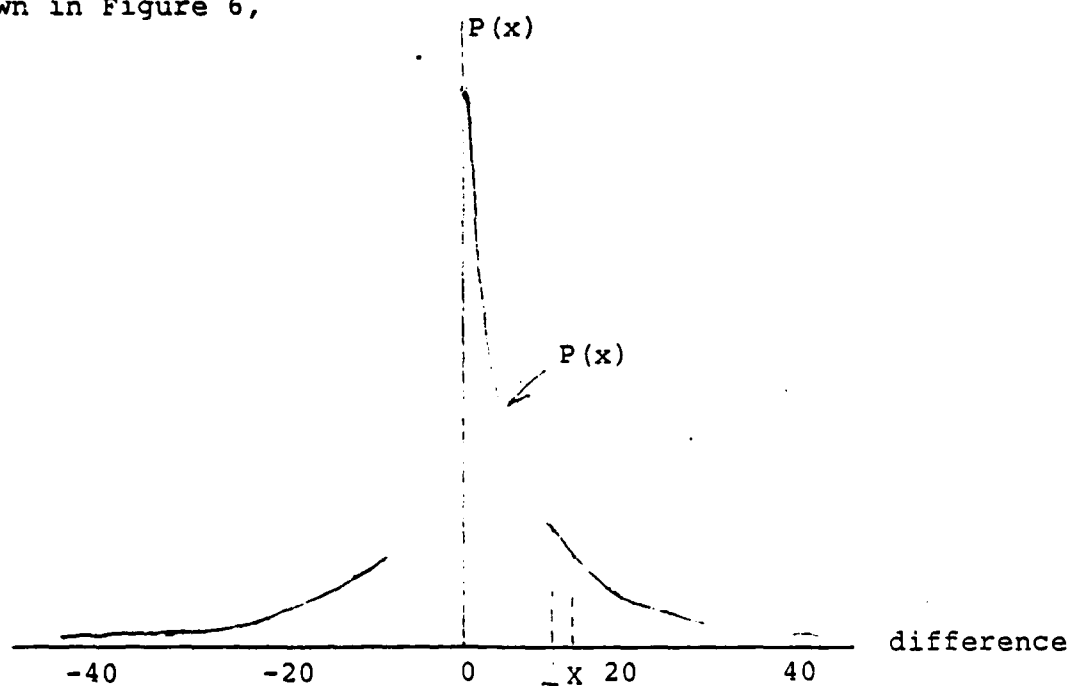
difference

-40   -20   0 X 20   40

Figure 6

$P(x)$ is the probability density function of differences falling into ΔX range. X is the value of a difference.

-- 10 --

the input range into a number of bins as illustrated in Fig. 5.
If an input falls into the Kth bin, the output is the value Q_1
corresponding to the center of Ith bin so that each input is
rounded off to the center of the bin into which it falls. A
uniform quantizer is one in which all bin widths are equal. Non-
uniform quantizers allow different bins to have different widths."

Let X represent any input value, and let Q_i be the corresponding
output of the quantizer. "If all values of X within the bins are
not equally likely then the squared error $(X_i - Q_j)^2$ must be
weighted by the probability density function $p(X_i)$ "[5].

$$P(X_i) = \lim_{\Delta x \to 0} \frac{F (X_i \le X \le X_i + \Delta X)}{\Delta X}$$

$F(X_i)$ represents the number of differences which have values
between $[X_i, X_i + \Delta X]$. If we choose $\Delta X = 1$ (because of digital
signal), $p(X_i)$ is approximately equal to the probability density
function.

In adaptive IDPCM the quantization strategy is to choose the
quantized levels (Q_i) so that they minimize the total quantizer
mean-square error. This error is defined as

$$e_{\bar{q}} = \sum_{i=1}^{n} \int_{x_i}^{x_i+1} (X - Q_i)^2 P(X) \, dx \qquad (3)$$

Taking a partial derivate of equation (3) with respect to
Q_i gives

$$\frac{\partial e_q}{\partial Q_i} = -2 \int_{x_i}^{x_i+1} (X - Q_i) P(X) dX = 0 \qquad (4)$$

$$Q_i = \frac{\int_{x_i}^{x_i+1} XP(X) dX}{\int_{x_i}^{x_i+1} P(X) dX} \qquad (5)$$

Since we have two unknown variables Q_i and X_i, we also take the partial derivative of equation (3) with respect to X_i

$$\frac{\partial e_q}{\partial X_i} = \frac{\partial}{\partial X_i} \left\{ \int_{x_i-1}^{x_i} (X-Q_{i-1})^2 P(X) dX + \int_{x_i}^{x_i+1} (X - Q_i)^2 P(X) dX \right\} = 0$$

and get $\quad (X_i - Q_{i-1})^2 P(X) - (X_i - Q_i)^2 P(X) = 0$

$$X_i - Q_{i-1} = Q_i - X_i \qquad (6)$$

From simultaneous equations (5) and (6)

$$Q_i = \frac{\displaystyle\int_{x_i}^{x_i+1} XP(X)\, dX}{\displaystyle\int_{x_i}^{x_i+1} p(X)\, dX} \qquad (5)$$

$$X_i - Q_{i-1} = Q_i - X_i \qquad (6)$$

optimal quantizer output levels are achieved, which is shown in Fig. 7
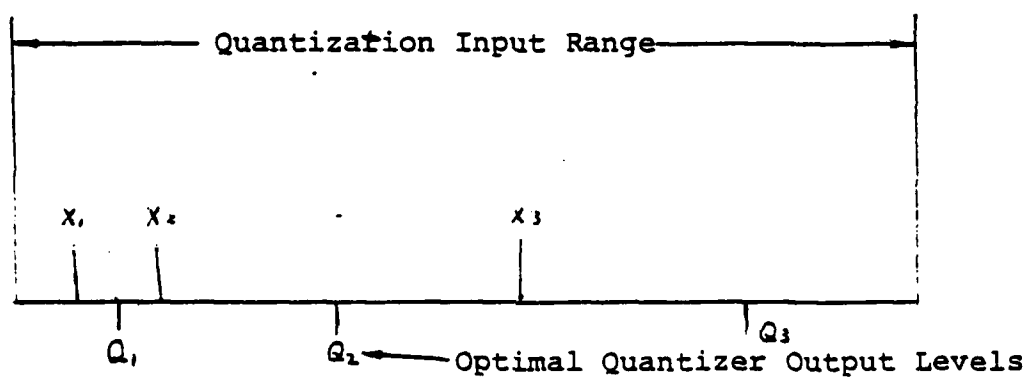

Fig. 7

Simulations of Adaptive Recursive IDPCM

In Fig. 7, the original image is sampled at 512x512 resolu-
tion with 8-bits per pixel. The image data compression steps were
as follows:

(1) The original image was subsampled at every 8th line and
every 8th pixel. Each subsample was quantized to 6-bits with a
uniform quantization. The maximum and minimum quantization levels
were 256 and 0.

(2) The middle point interpolated pixels were subtracted from
the original pixels, and the differences were optimally quantized
in 3 bits if the mean-square differences in a subimage were larger
than 4.0, (the average error in subsamples caused by quantizing
8-bit original pixels to 6-bit subsamples). Of course, those
differer. es were not stored if the mean-square differences in a
subimage were smaller than 4.0.

(3) The subsamples and the quantized middle point differences
were then used to calculate the second set differences. Suitable
thresholds for the mean-square differences in a subimage were
chosen to determine how many bits are being used for quantizations.

$$N_d = \begin{cases} 0 & \text{if } R' \varepsilon\ C_s \\ 2 & \text{if } R' \varepsilon\ C_m \\ 3 & \text{if } R' \varepsilon\ C_c \end{cases}$$

C_s, C_m, and C_c mean simple-detail, medium-detail, and complex-
detail subimages.

(4) The subsamples and quantized differences according to the
R' were used to reconstruch the image. Fig. 8 shows the original

image, Fig. 9 shows the processed image by using recursive IDPCM, and Fig. 10 shows the image by using adaptive recursive IDPCM. Comparing Fig. 9 with Fig. 10, we can see that the details in Fig. 10 did improve.

The total bit requirements for the images is the sum of bits for subsamples, differences, and the modes of each subimage. Since the number of bits for representing modes of each subimage is very small, these extra bits can be negligible.

total bits = $3 \times 3cN_1 + 6 \times N_2 + 2 \times 2 \times N_3 + 3 \times 12 \times N_4$

N_1 = number of subimages of R' > 4.0, N_2 = number of total subimages, N_3 = number of subimages of its R' (the second set differences) between 4.0 and 12.0, N_4 = number of subimages of R' (the second set differences) >12.0. In adaptive recursive IDPCM for the image shown:

> Bit Rate of Per Pixel = 0.35259
>
> Mean-Square Error = 0.001798

Original Image

Fig. 8

Recursive IDPCM BPP=0.3577296 , MSE=0.0015513

Fig. 9

Adaptive Recursive IDPCM BPP=0.35259, MSE=0.001798

Fig. 10

$R' \leqslant 4.0$

Fig. 11

$4.0 < R' \leqslant 12.0$

Fig. 12

$R' \geq 12.0$

Fig. 13

References

[1] B. R. Hunt and Zhidao Cao, "Image Data Compression By Recursive IDPCM", to be published.

[2] B. R. Hunt, "An optical analogy to DPCM digital image compression", SPIE Processing, Vol. 119, 'Applications of Digital Image Processing', PP. 85-90.

[3] Alan V. Oppenheim, Editor, "Applications of Digital Signal Processing", Chapter 4, "Digital Image Processing".

[4] Two-Dimensional image coding by Micro-Adaptive Pictyre Sequencing(MAPS), Vol. 119, SPIE 1977.

[5] Digital Image Processing, Rafael C. Gonzalez.

[6] Digital Image Processing, William K. Pratt.

[7] C. C. Cutler,"Differential Quantization of Communication Signals" Patent2-605-361, Application June 1950, Issuance July 1952.

[8] R. E. Graham, "Predictive Quantizing of Telecision Signsls" IRE WESCON Convention Record, Part 4, 1958, 142-157.

[9] J. O. Limb, "Adaptive Encoding of Picture Signal", in Picture Bandwidth Compression, T. S. Hyabg and O. J. Tretiak, Eds., Gordon and Breach, New York, 1972, 341-382.

[10] Anil K. Jain, "Image Data Compression: A Review", Proceedings of IEEE, vol. 69, No. 3, pp. 349-389.

(V.) Applications of Stationary Transform Processing in
Space-Variant Image Processing

This work has focused on three areas: (1) Improved estimates
of local autocovariance statistics for direct implementation of
spatially-adaptive image processing algorithms. (2) Development
of adaptive image restoration algorithms for nonstationary images
degraded by blur and additive noise. (3) Development of adaptive
image enhancement algorithms for local contrast transformation,
with application to color imagery.

The results of this work are reported in the sections to
follow. In addition, Appendix A gives details of an extension
to the geometric-transform algorithm (see 1982 and 1983 AFOSR
Annual Reports). This computer algorithm is capable of generating
a grid whose cells are of specified area and unspecified shape.
The procedure is iterative, producing a better solution after each
iteration, and converging even for quite large differences in
cell area.

Improved Estimates of Local Autocovariance Functions

Our previous work in this area used the first-order Markov
model for autocovariance functions estimated from image subblocks.
First-order parameters were obtained by fitting an exponential
curve to the autocovariance data. Recent tests, however, indicate
that this approach gives highly variant statistical estimates, due
to the inclusion of unreliable high-order lag data in the curve
fitting process.

In Appendix B we report results from our experiments to find useful measures of local image autocovariance parameters from small subblocks of data. The basis of this work is a new technique for estimating the correlation parameters of first-order Markov (nonseparable) exponential autocovariance models. The method assumes that image data is stationary within NxN-pixel subblocks. A measure of the usefulness of the ensuing correlation parameters may be had by observing their correlation with signal activity within the scene. Subblocks of dimension N=16 are shown to provide estimates fitting this criteria, even when the original data is degraded by significant amounts of blur and noise. This latter fact bodes well for spatially-adaptive image restoration applications. An application to block-interpolative data compression is also included in Appendix B.

Adaptive Image Restoration

The basic concept of stationary transforms in image restoration is as follows:

. Assuming the image to be degraded by blur and additive noise, then -

. Estimate the autocovariance parameters by the method outlined previously.

. Apply geometric transforms to produce stationarity in correlation length.

. Filter the resulting data with, e.g., a Wiener filter based on the stationary autocovariance distribution.

. Apply the inverse geometric transform to restore the original geometry.

This approach is quite feasible for 1-D data processing, since the geometric transforms simply translate into straightforward decimation and interpolation operations. However, we showed in the 1983 AFOSR Annual Report that the 2-D implementation of the above steps results in highly distracting artifacts in the form of regular patterns of correlated noise. In an attempt to overcome this limitation we investigated a modification of the above technique:

. Estimate autocovariance parameters in NxN-pixel subblocks as before.

. Subsample (decimate) subblocks by a linear scaling factor proportional to these parameters in order to produce constancy over all the parameter set.

. Place the new subblocks at the centers of the original subblocks, and apply a nonadaptive Wiener filter as before.

. Restore the subblocks to their original dimensions by interpolation.

Figures 1(a) and 1(b) show the results of this procedure for the case where additive noise is the only source of degradation. The resulting restoration in Figure 1(b) is encouraging, since the residual blocking effect can most likely be removed by bandpass filtering incorporated at the same time as the noise-filtering.

We have also investigated several new approaches for adaptive image restoration using space-varying filters. A simple type of

space-variant deblurring is implemented via an Adaptive Biased Laplacian operator. This is a 3x3 kernel given by:

$$L(x,y) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8+d(x,y) & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (1)$$

The deblurring is given by a convolution between $L(x,y)$ and the image $i(x,y)$. The term $d(x,y)$ is a space-varying bias which adapts to local image statistics; for example, image variance $var(x,y)$-

$$d(x,y) = f(var(x,y)) \quad (2)$$

where $f(\ )$ is monitonically decreasing.

When $var(x,y)$ is high, for example at large edges we want $L(x,y)$ to be a high-pass filter. In regions of low variance, we want $L9x,y)$ to be all-pass. These constraints are satisfied when

$$d(x,y) = 1 + 9x_{exp}(-fxvar(x,y)^2) \ ; \ f>0 \quad (3)$$

From eqn. (3), we notice that the minimum value of $d(x,y)$ is unity, and the maximum is 10. In conjunction with f=0.011, this range is found to give good results.

Normalization of $L(x,y)$ is necessary to avoid large spatial gain variations during the convolution: the resulting normalized kernel is given by;

$$L'(x,y) = L(x,y) \div d(x,y) \quad (4)$$

The adaptively-processed image is,

$$g(x,y) = i(x,y) \ ** \ L'(x,y) \quad (5)$$

Separating terms in equ. (5), we have;

$$g(x,y) = i(x,y) ** (L/d(x,y) + i(x,y) \qquad (6)$$

where L is the usual 3x3 Laplacian kernel consisting of -1 in all elements except the center element of 8. An additional weight in the first term prevents saturation, i.e.,

$$g(x,y) = 0.3[i(x,y) ** (L/d(x,y))] + i(x,y) \qquad (7)$$

Figures 2(a) and 2(b) show an image before and after processing with equ. (7). The effect of unwanted noise boosting in areas of low signal activity is bypassed in this adaptive approach.

Adaptive Image Enhancement

A new algorithm for adaptive contrast enhancement is described in this section. The aim of the algorithm is to compensate for nonuniform illumination in scenes, and is based on the often observed fact that local maps of mean and variance correlate quite strongly in the positive sense. The philosophy of the algorithm may be outlined as follows:

. If the local mean $\bar{i}(x,y)$ is high, then leave the image unaltered.

. If the local mean $\bar{i}(x,y)$ is low, then increase the variations about the mean as well as the mean itself.

For convenience we will drop the (x,y) image coordinates and represent the algorithm as follows:

$$g = k_1(i - \bar{i}) + k_2 + \bar{i} \qquad (8)$$

The terms k_1 and k_2 are functions of $\bar{i}(x,y)$. In order to satisfy

the algorithm specifications they should exist within the following limits:

$$k_1 = < 1 \; ; \; \overline{I} = 255$$
$$< 3 \; ; \; \overline{I} = 0$$
$$\max(k_1) = 3$$
$$k_2 = < 0 \; ; \; \overline{I} = 255 \qquad (9)$$
$$< C \; ; \; \overline{I} = 0$$

Functional forms which accomodate these limits are given in a general form by,

$$k_1 = A(\frac{1}{\overline{I}} - \frac{1}{255}) + 1 \qquad (10)$$

$$k_2 = C(1 - \frac{\overline{I}}{255})^t \qquad (11)$$

Optimum values for the constants based on experimental trials are: $A = 100$, $C = 70$, $t = 2$.

The terms k_1 and k_2 are, respectively, the local contrast and local mean adjustment factors. Calculation of local means $\overline{I}$ is achieved by neighborhood averaging over 9x9-pixel windows.

Figures 3(a) & (b) illustrate the effect of this adaptive image enhancement. Individual red, green, and blue bands were processed separately to produce the color imagery.

Note

Appendix A was submitted for publication in "Computers and Graphics", January 1984.

Appendix B was submitted for publication in IEEE Trans. on

<u>Acoustics, Speech and Signal Processing</u>, February 1984.
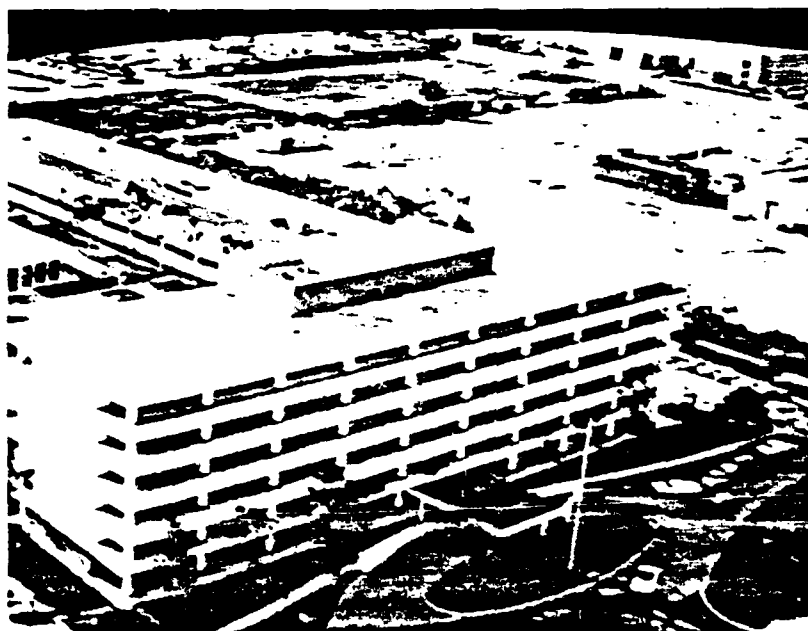
Acknowledgement

(a) Image degraded by additive
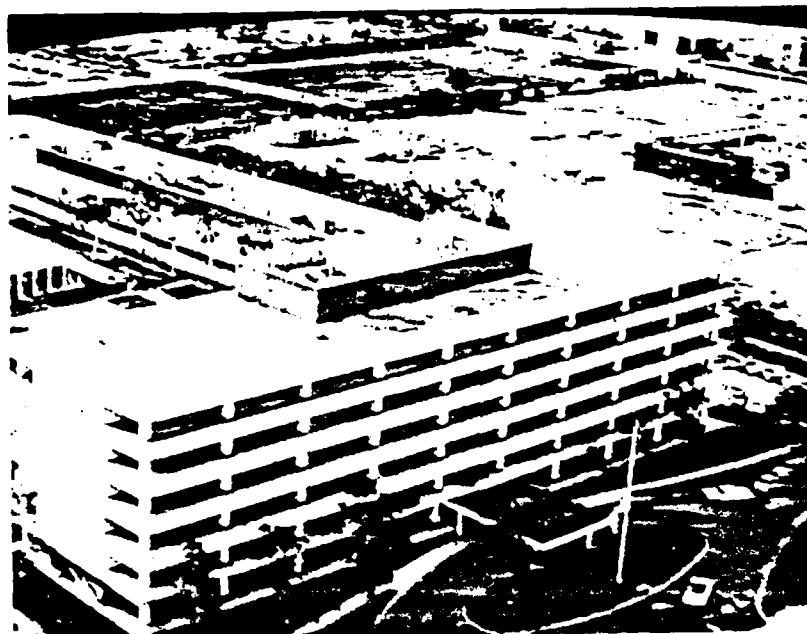Gaussian noise (standard
deviation = 20)

(b) Restoration

Figure 1. Block-adaptive image restoration.

(a) Original image.

(b) Adaptive restoration using eqn.(7)

Figure 2. Adaptive image restoration.

(a) Original image.

(b) Adaptive image enhancement.
(vertical stripes are scanner artifacts).

Figure 1. Adaptive image enhancement

APPENDIX A

An Algorithm for Generating Connected
Quadrilaterals of Specified Area

I. INTRODUCTION.

This paper addresses a simply-posed problem: starting with a simple grid of square cells, determine the coordinates of a new grid in which the area of every cell is specified individually. The shapes of cells in the new grid are unspecified.

This problem arose in an image processing/remote sensing application [1]. We envisage that the solution may find other applications; for example, in computer graphics, mapping, integrated circuit fabrication...

The flavor of the paper is heuristic. Convergence of the algorithm is demonstrated by examples, without proof.

II. ALGORITHM.

Figure 1 illustrates the broad objectives and constraints of the problem. Each unit-area cell in the original regular grid is to be expanded to form an arbitrary shaped quadrilateral of specified area. No other constraints are imposed, apart from the obvious ones of contiguity and connectivity in the transformed grid. Simply stated, we wish to know the coordinates of each cell in the transformed grid, relative to an arbitrary set of orthogonal axes in the x-y plane. If we define a regular grid of control points whose coordinates lie at the vertices of each cell, then the problem is to determine how such a regular grid is distorted by the many interactions of local area expansions centered on each cell.

Our approach is to superpose the expansion of each cell on the global grid, subject to the constraint that each expansion should minimally affect the relative geometry of other cells. This philosophy is easily explained by

-1-

a 1-D illustration. In Figure 2(a) we begin with uniformly spaced control
points. In Figure 2(b) an expansion factor of 2 is applied to the center
of a cell at x_0, resulting in localized stretching. The next expansion
is applied to the neighboring cell, and so on. Continuing this process
yields a 1-D control point array whose coordinates reflect the desired ex-
pansion of cells. No iterations are required. The basic 1-D geometric
transformation can be represented by a spatial displacement Δx,

$$\Delta \dot{x} = (E - 1) \frac{d}{2} \tag{1a}$$

where E is an "expansion factor," and

$$\text{Compression} \triangleq 0 \leq E < 1$$

$$\text{Expansion} \triangleq 1 \leq E < \infty \tag{1b}$$

In the 2-D case, it is not possible to design a transformation which
expands the area of a single cell, while preserving the geometry of all
other cells. What is needed is a technique for reducing the mutual interference
of local cell expansions. Consider a single radial expansion centered on a
single cell in a grid of control points, as shown in Figure 3. Straight-
forward radial expansion takes place within a circle of radius r centered
on the expansion cell. Elsewhere, more remote control points are simply dis-
placed. Hence, we can define a geometric transformation as follows:

$$\left. \begin{array}{l} \Delta x = x(E-1) \\ \Delta y = y(E-1) \end{array} \right\} \quad (x^2 + y^2)^{1/2} < r \tag{2a}$$

$$\left.\begin{array}{l} \Delta x = r \cdot \cos \theta \cdot (E-1) \\ \Delta y = r \cdot \sin \theta \cdot (E-1) \end{array}\right\} \quad (x^2 + y^2)^{1/2} \geq r \qquad (2b)$$

where x, y, and θ are control point coordinates relative to the center of expansion (see Figure 3). Nominally, $r = d/\sqrt{2}$ in order to confine expansion to a single cell as much as possible. The result of applying a linear expansion of $E = 2$ to a single cell is shown in Figure 4. Note that the area expansion of the cell is E^2, or 4. Also, notice that the influence of this expansion on other cells diminishes at greater distances from the expansion center.

We have found that the simple geometric transformation in Eqn. (2) works well if applied sequentially to all cells in the grid, provided that correction factors are used to compensate for the influence of single-cell expansions on all other cells. Furthermore, iterating the procedure yields the desired cell areas in most cases.

The iterative algorithm proceeds as follows:

Notation

Assume initially an $n \times m$ grid of unit-area cells.

C_{ij} – cell (i,j) $(1 \leq i \leq n : 1 \leq j \leq m)$

E_{ij} – desired linear expansion of C_{ij}

E_{ij} – desired area of C_{ij}

A_{ij} – actual area of C_{ij} at any time.

Algorithm:

```
┌─START
│
├────── For i = IORDER (1)  →  IORDER (n)
│
│           For j = JORDER (1)  → JORDER (m)
│
│           Measure A_{ij}
│
│           Form the corrected expansion E = E_{ij}/√A_{ij}
│
│           Apply Eqn. (2) to C_{ij}, using E.
└─
         END OF LOOPS


         RETURN TO START, repeat for K iterations.
```

The arrays IORDER and JORDER are needed to generalize the order in which the cells are treated, in other words, each cell is considered only once during an iteration, but cells may be addressed in any desired sequence. We will refer to this as the cell sequence.

NOTE: In applying Eqn. (2), we measure x and y from the specified control point to the centroid of C_{ij}.

III. EXAMPLES*

Most of our examples concern a grid of 16 cells; i.e., 25 control points. The arbitrary set of expansions shown in Figure 6 is used throughout the tests. Several variations of the iterative algorithm are presented in order to demonstrate the following:

- effect of cell sequence.

- effect of radius r in Eqn. (2).

- existence of an upper limit on E_{ij}.

---

* The figures in this section are not drawn to scale.

In all tests, we employ an rms expansion error measure for the solution, given by:

$$e(k) = \left\{ \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} [E_{ij}^2 - A_{ij}]^2 \right\}^{1/2} \tag{3}$$

where k is the iteration index.

Cell Sequence.

Figure 5 shows the notation for cell sequence. Tests indicate that row-by-row or column-by-column sequences are superior to, say, circular or spiral ordering. Indeed, for the particular 16-cell example selected, only the former yield convergence in the solutions. Figures 6(a) and (b) show the transformed cells after 100 iterations using the following cell sequences.

(i) Fig. 6(a) - 1 2 3 4, 5 6 7 8, 9 10 11 12, 13 14 15 16.

(ii) Fig. 6(b) - 1 2 3 4, 8 7 6 5, 9 10 11 12, 16 15 14 13.

In both (i) and (ii), the nominal value $r = d/\sqrt{2}$ was used. Evidently, the rate of convergence and form of solution is affected by the cell sequence, although only slightly in this example. However, these effects depend greatly on the expansion values. In some cases, even the sequence in (ii) does not yield a solution.

Radius parameter, r.

The nominal value of $r = d/\sqrt{2}$ is not optimum. For most expansion sets, a value can be found which dramatically increases the convergence rate. For example, a value of $r = \sqrt{2}d$ in conjunction with the cell sequence in (ii) gives a solution with $e(100) = 7 \times 10^{-6}$; i.e., a factor of 100 better than the nominal r solution. The resulting grid of cells is shown in Figure 6(c). Clearly, this value of r also leads to a somewhat different solution.

Upper limit on expansions.

The series of results in Figures 6(d),(e), and (f) illustrates the behavior of the algorithm as the expansion applied to one particular cell is increased. Ultimately, the boundaries of adjacent cells cross, creating additional cells (cell nos. 15 and 16 in the example shown). Our experience is that the maximum allowable expansion(s) is a function of r and the cell sequence.

Some additional interesting results are now discussed. Tests of the iterative algorithm using symmetrical expansion sets yield Figures 7(a) and (b). We observe that the asymmetry of the cell sequence is responsible for the asymmetry of the solutions. An example using 64 cells is given in Figure 8.

IV. REMARKS.

We have demonstrated the behavior of an iterative procedure for generating connected quadrilaterals of specified area. The solutions are nonunique, and depend on two free parameters - radius r in Eqn. (2), and the cell sequence. Obviously, the nonuniqueness is a consequence of the overdeterminacy of the problem. Additional constraints on the quadrilateral cells, such as shape constraints, would reduce the number of solutions. The algorithm presented here usually converges, even when the expansion set spans a large dynamic range.

-6-

ACKNOWLEDGEMENT

REFERENCES

[1] R. N. Strickland, "Transforming Images into Block Stationary Behavior," Applied Optics, Vol. 22, No. 10, pp. 1462-1473, May, 1983.

Unit-area cells   Cells of specified area
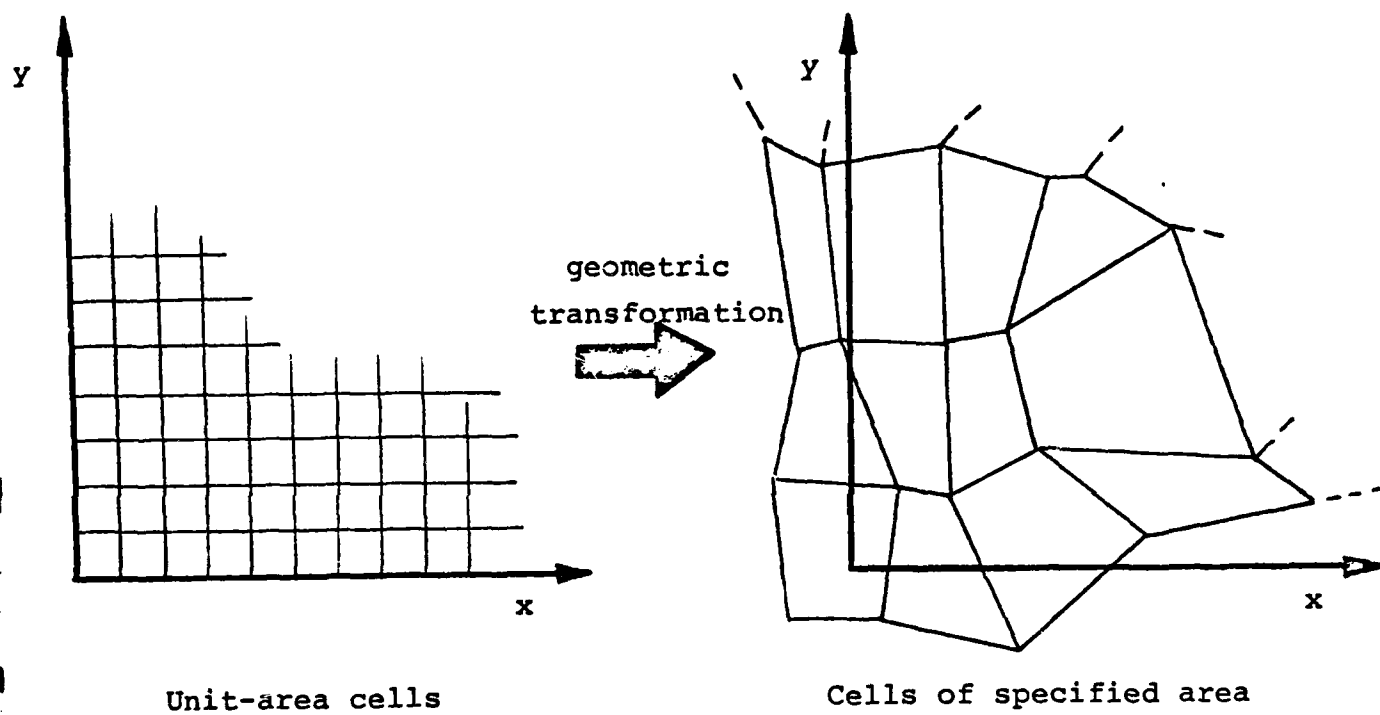
Fig.1   Objectives
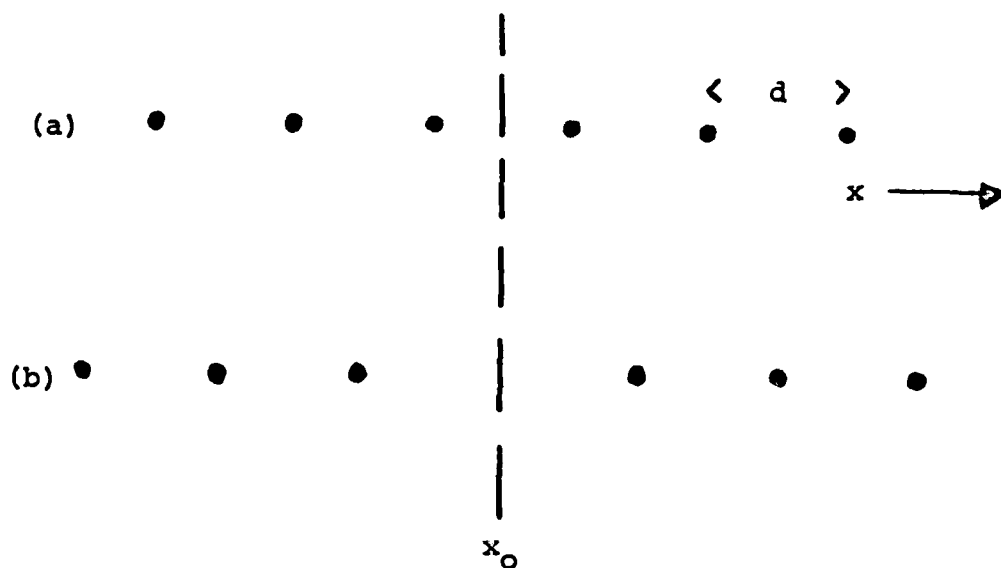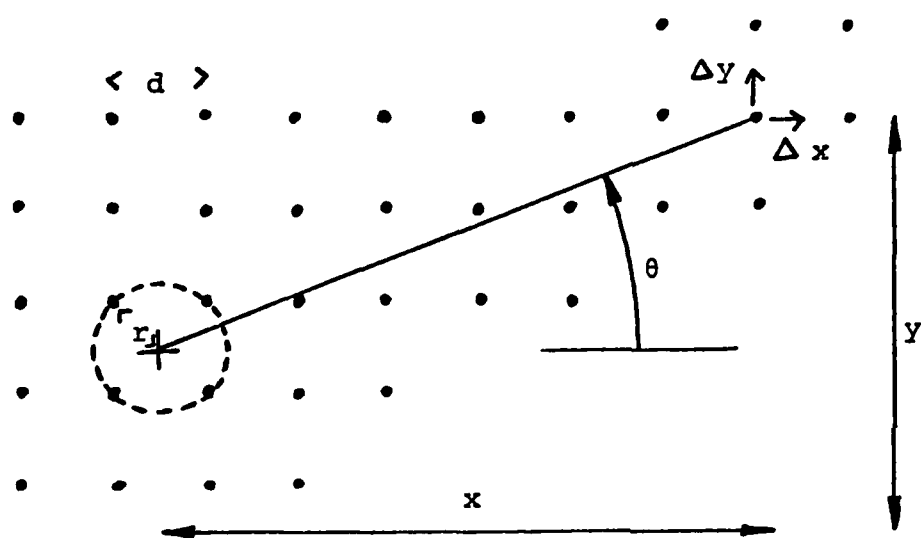
Fig.2   1-D illustration of cell expansion

Fig.3   2-D   Expansion geometry
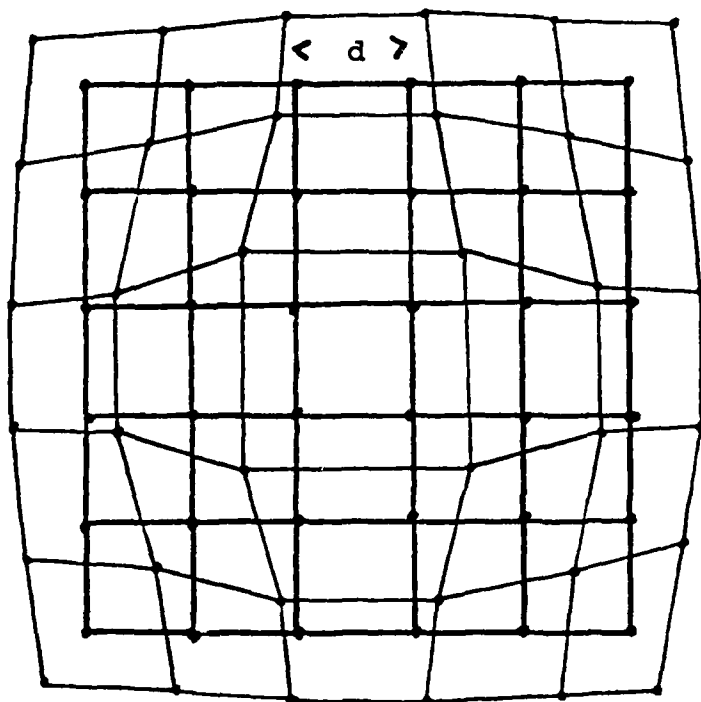
Fig.4 Expansion applied to the center cell of a
5 x 5 grid using Eqs.2(a),(b). $(r=d/\sqrt{2})$

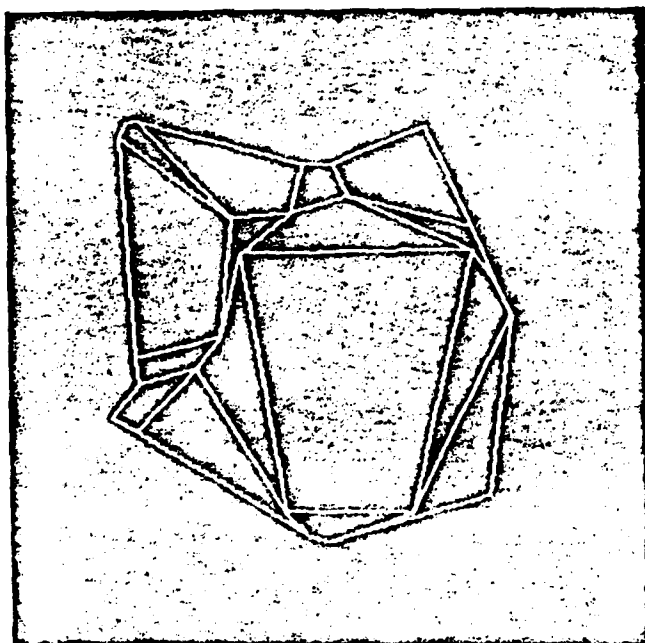|			
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig.5   Cell sequence notation
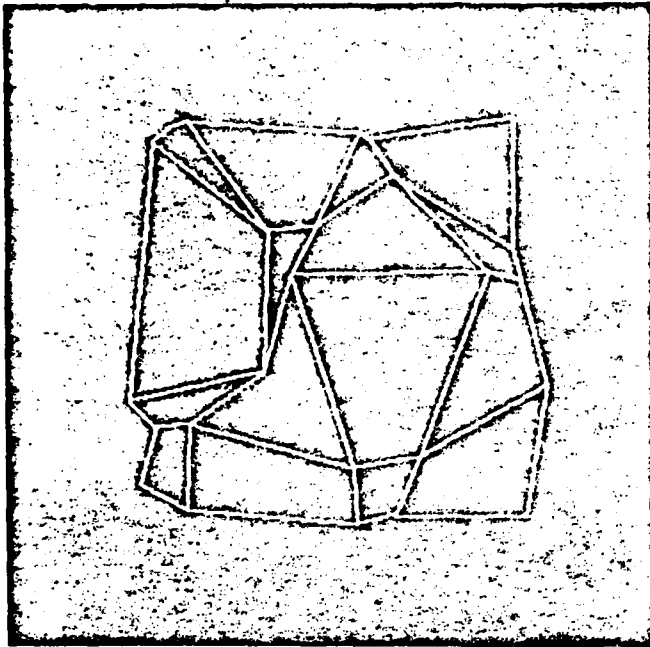
Figure 6(a): $r = d/\sqrt{2}$; cell sequence (i). $e(100) = 8 \times 10^{-4}$.
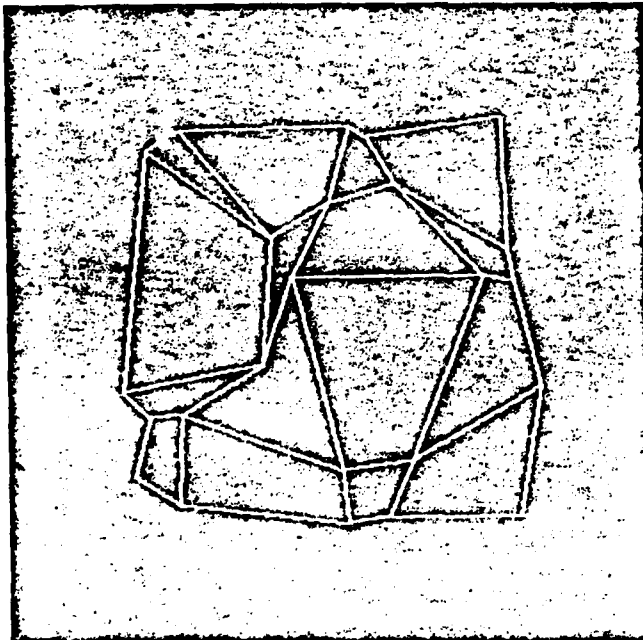
Figure 6(b): $r = d/\sqrt{2}$; cell sequence (ii). $e(100) = 7 \times 10^{-4}$.

Figure 6: Results of 100 iterations of the algorithm.

Expansions
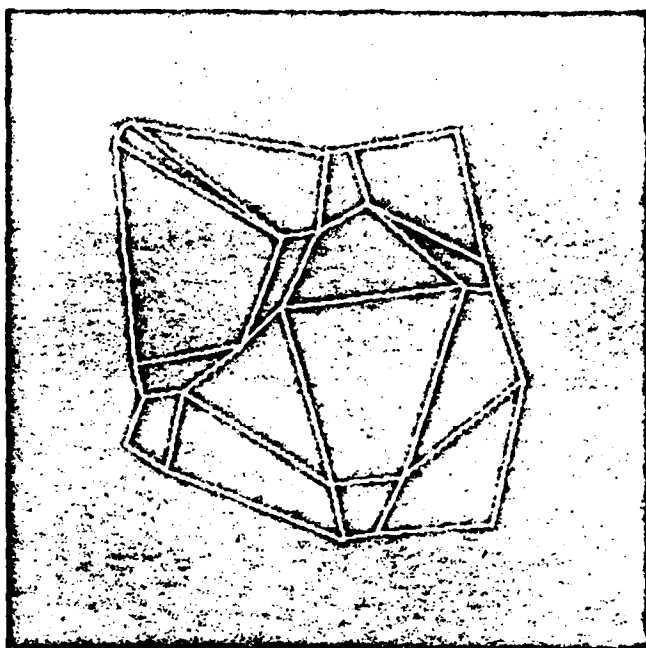
Figure 6(c): $r = \sqrt{2}\, d$; cell sequence (ii). $e(100) = 7 \times 10^{-6}$.

Figure 6(d): $r = \sqrt{2}\, d$; cell sequence (ii). $E_{33} = 5.0$.
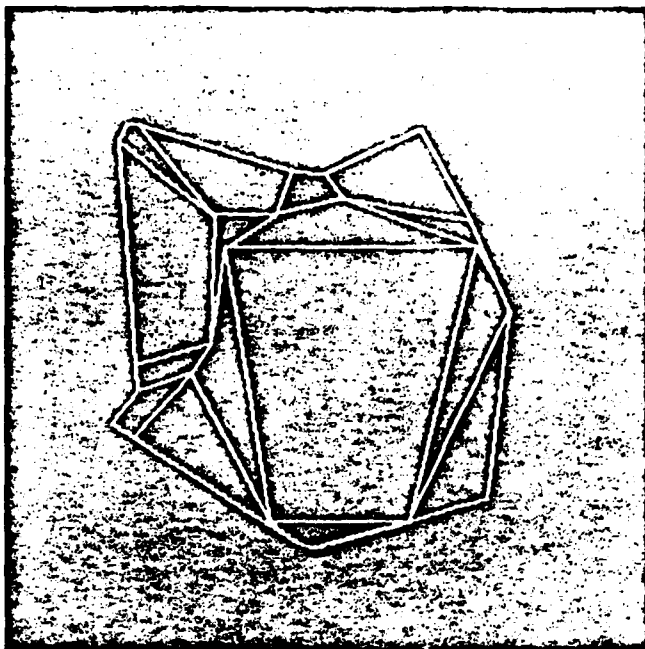$e(100) = 4 \times 10^{-4}$.

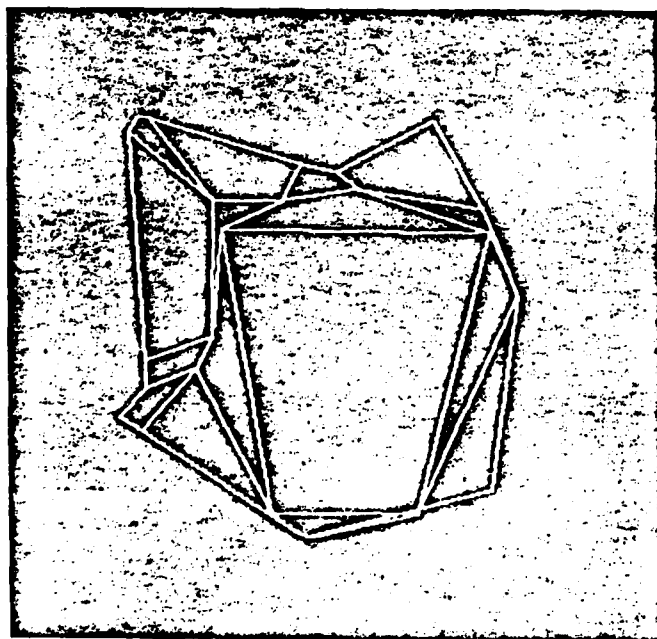Figure 6(e): $r = \sqrt{2}\ d$; cell sequence (ii). $E_{33} = 5.5$. $e(100) = 3 \times 10^{-4}$.

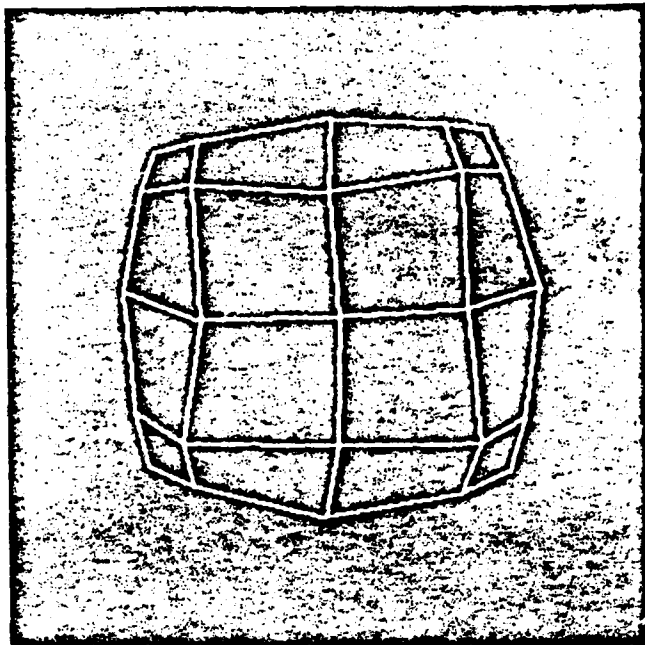Figure 6(f): $r = \sqrt{2}\ d$; cell sequence (ii). $E_{33} = 6.0$. $e(100) = 10^{-3}$.

Figure 7(a): $r = \sqrt{2}\,d$; cell sequence (i). $e(100) = 6 \times 10^{-6}$.

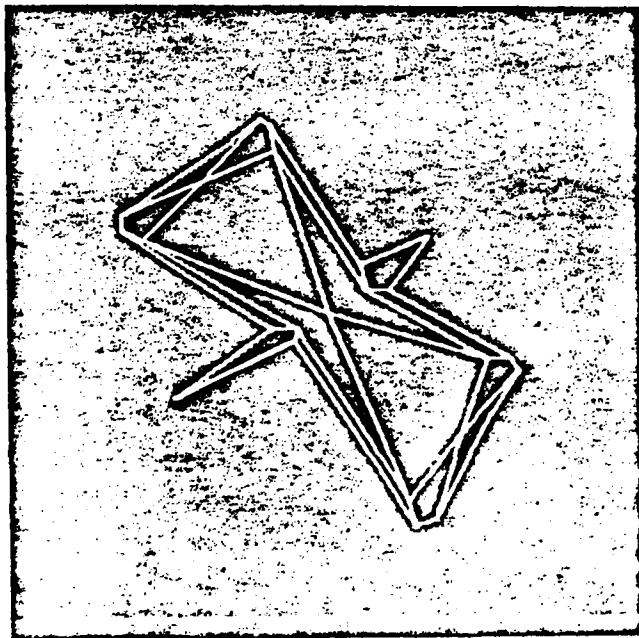Expansions: 1 2 2 1, 2 3 3 2, 2 3 3 2, 1 2 2 1.

Figure 7(b): $r = \sqrt{2}\,d$; cell sequence (i). $e(100) = 2 \times 10^{-6}$.
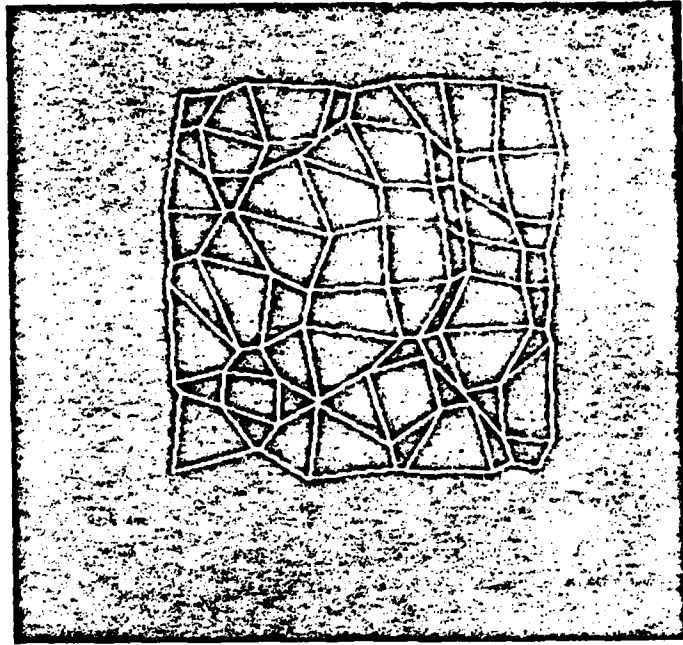
Expansions: 1 1 1 1, 1 3 2 1, 1 2 3 1, 1 1 1 1.

Fig.8: Result using 8 x 8 grid of expansions
arbitrarily chosen from 1.0, 1.5, 2.0 .
(e(50) = 0.005)

APPENDIX B

Estimation of Local Statistics for Digital
Processing of Nonstationary Images

I. INTRODUCTION.

The design of digital image processing algorithms is critically dependent on the spatial statistics of the imagery. The largest class of algorithms assumes stationary statistics; for example, the Wiener filter for least-squares image restoration [1], and the DPCM coder in data compression [2] . The main advantage of such algorithms lies in the relative simplicity of their design and implementation. A drawback, however, is that images are inherently nonstationary; consequently, the results of processing are suboptimal. The Wiener filter, for example, is designed from estimates of the object (and noise) autocovariance functions. The question is: how do we obtain meaningful estimates of the object covariance function? Typical imagery consists of large regions with nearly uniform intensity, plus a relatively few small areas of high signal activity, such as edges, natural texture, etc. Therefore, an estimate of the autocovariance based on the global image data would yield a rather broad function, since most of the data tends to be highly correlated. A Wiener filter designed from such an estimate would yield a very smooth restoration, and important edge detail would remain blurred. Since the Wiener filter provides a compromise between noise smoothing and deblurring, with a fairly conservative bias towards noise smoothing, a better approach is to estimate autocovariance from regions of low correlation, i.e. edges, etc. The resulting restored images exhibit a greater degree of sharpness, although they are inevitably noisier than normal least-squares restorations. Nevertheless, all of these techniques are handicapped because they do not account for nonstationarity in the image data.

A more recent class of image processing algorithms accommodates this nonstationarity. These algorithms share the characteristic of adapting to, or tracking, the local variations in image correlation. The resulting processors are space-varying, in accordance with the image nonstationarity.

Lahart [3] has reported a method of implementing local least-squares resto-
ration in which two different autocovariance functions were used, depending
on whether pixels were considered to belong to high or low signal-activity
categories. Anderson and Netravali [4] used a masking function to implement
a form of space-varying noise-smoothing which adapted to local signal details.
A different approach to achieving the same goal is due to Kasturi [5], who
considered the particular case of signal-dependent noise.(See also Froehlich,
Walkup, and Asher [6].) The benefits of space-varying processing have proved
even more dramatic in the field of image data compression. A review of adapt-
ive coding methods is provided by Habibi [7]. All of the work cited above
depends on some kind of direct estimation of local image statistics. Other
methods use nonstationary models indirectly; an excellent example is Widrow's
LMS adaptive filter [8].

In order to quantify the nonstationarity of images, and, often, in order
to implement adaptive processing, we require estimates or measurements of
the local image statistics, specifically the autocovariance function. The
simplest way of achieving this is to divide the image into N x N - pixel
subblocks ($N \approx 16$), and calculate the standard biased or unbiased autocovar-
iance function of each subblock. In effect, each subblock is treated as part
of a wide-sense stationary field. It is well-known, however, that reliable
power spectral estimation requires much larger amounts of data. Neverthe-
less, as our work shows, it is possible to obtain useful maps of local auto-
covariance parameters if we assume simple parametric autocovariance models.
Specifically, we employ the popular first-order Markov random field model
for image data. Three topics are addressed herein:

- estimation of local autocovariance parameters.

- optimum choice of N.

- correlation between the estimated autocovariance
  parameters and observed signal activity.

In the sections to follow, we review the autocovariance model and discuss possible ways of estimating the parameters of each local subblock. Examples are given which illustrate some of the problems encountered when the estimates are applied to real image data. Finally, we give an example of the value of local estimates in adaptive data compression.

II. LOCAL IMAGE MODELS.

Global wide-sense stationary image data is generally modeled by a first-order Markov random field with a nonseparable exponential autocovariance function [9] given by

$$C(k, \ell) = \sigma^2 \exp \left\{ - \sqrt{\rho_1 k^2 + \rho_2 \ell^2} \right\} \tag{1}$$

where σ^2 is the variance. Typical values adopted for the correlation parameters are $\rho_1 = \rho_2 = 0.0025$, resulting in the simpler isotropic form

$$C(k, \ell) = \sigma^2 \, 0.95^{\sqrt{k^2 + \ell^2}} \tag{2}$$

This model is often assumed in nonadaptive DPCM coding systems.

A new model for nonstationary image data $f(i, j)$ is given by:

$$C(k, \ell : i, j) = \sigma^2(i, j) \exp \left\{ - \sqrt{\rho_1(i, j) k^2 + \rho_2(i, j) \ell^2} \right\} \tag{3}$$

It is assumed that $f(i,j)$ is stationary over any $N \times N$ - pixel subblock centered on the coordinates (i,j), and that values of σ^2, ρ_1, and ρ_2 may be assigned to every location (i,j). (In practice, these values will be correlated with their neighbors. Later in the paper we will discuss the implications of this.) The next section discusses procedures for estimating the spatially-variant autocovariance parameters $\rho_1(i,j)$ and $\rho_2(i,j)$.

III. ESTIMATION OF LOCAL CORRELATION PARAMETERS.

The starting point for our estimates of ρ values is the usual biased autocorrelation estimate,

$$\hat{R}(k,\ell:i,j) = \frac{1}{N^2} \sum_{m=i-\frac{N}{2}}^{i+\frac{N}{2}-|k|-1} \sum_{n=j-\frac{N}{2}}^{j+\frac{N}{2}-|\ell|-1} f(m,n)\, f(m+k,\, n+\ell) \tag{4}$$

or the biased autocovariance estimate, given by,

$$\hat{C}(k,\ell:i,j) = \frac{1}{N^2} \sum_{m=i-\frac{N}{2}}^{i+\frac{N}{2}-|k|-1} \sum_{n=j-\frac{N}{2}}^{j+\frac{N}{2}-|\ell|-1} [f(m,n) - \bar{f}(i,j)] \tag{5}$$

$$[f(m+k,\, n+\ell) - \bar{f}(i,j)]$$

where $\bar{f}(i,j)$ is the spatial mean of an $N \times N$ - pixel subblock centered on (i,j).

Simple parametric estimates.

From Eqn. (3), simple parametric estimates of ρ_1 and ρ_2 are given by:

$$\hat{\rho}_1(i,j) = \left\{ \frac{1}{(k_2 - k_1)} \, \ell n \left[\frac{\hat{C}(k_1,0:i,j)}{\hat{C}(k_2,0:i,j)} \right] \right\}^2$$

$$\hat{\rho}_2(i,j) = \left\{ \frac{1}{(k_2 - k_1)} \, \ell n \left[\frac{\hat{C}(0,\ell_1:i,j)}{\hat{C}(0,\ell_2:i,j)} \right] \right\}^2$$

(6)

When $k_1 = 0$ and $k_2 = 1$, these estimates are equivalent to the first-order autoregressive (AR) estimates found in Box-Jenkins time series analysis [10]. For example, ρ_1 is then

$$\hat{\rho}_1(i,j) = \left\{ \ell n \left[\frac{\hat{C}(0,0:i,j)}{\hat{C}(1,0:i,j)} \right] \right\}^2$$

(7)

In practice, our constraint that N should be small results in autocovariance data with a high degree of variance, causing the ρ-estimates to be inaccurate. Furthermore, the values of $\hat{C}(1,0:i,j)$ and $\hat{C}(0,1:i,j)$ can be (and often are) negative, rendering Eqns. (6) and (7) indeterminate. For example, Figure 2 shows a map of $\hat{\rho}_1(i,j)$ obtained by applying Eqn. (6) to the image in Figure 1 with $k_1 = 1$ and $k_2 = 3$. The map was obtained by computing $\hat{\rho}_1$ at every pixel coordinate in the original image, with overlapping subblocks of 16 x 16 pixels. The large dark areas correspond to negativity in the argument of ℓn in Eqn. (6), caused by negative values of either $\hat{C}(k_1,0:i,j)$ or $\hat{C}(k_2,0:i,j)$.

At this point we should note the influence of the choice of lags k_1 and k_2 on the estimation quality. Estimation theory tells us that the lowest order lags provide the most reliable autocovariance data (least variance) [11],

suggesting that $k_1 = 0$, $k_2 = 1$ be the optimum data to use. However, the autovariance at zero lag often contains a significant "error" equal to the variance of unwanted film-grain noise (assuming the image data originates from film). This explains the choice of k_1 above. The selection of k_2 involves a tradeoff between the greater reliability of low order lag data and the increased error sensitivity of Eqn. (6) when k_2 is small. However, this form of estimation gives poor results for any combination of lag data, as Figure 2 demonstrates.

Modified parametric estimates.

A new method of ρ-estimation avoids the indeterminacy problem encountered in the simple estimation of Eqn. (6). For discussion purposes, estimation is executed in two stages, although the calculations can be combined in a single step.

First, the image is radiometrically transformed [12],[13] to yield data exhibiting block-stationary mean and variance. The transform is given by,

$$g(i,j) = \frac{\sigma_s}{\sigma(i,j)} [f(i,j) - \overline{f}(i,j)] + \overline{f}_s \qquad (8)$$

where $\sigma(i,j,)$ and $\overline{f}(i,j)$ are measured in an M x M – pixel subblock centered on (i,j). The mean and variance measured over any M x M – pixel subblock of $g(i,j)$ are the constants $\overline{f}_s$ and σ_s^2, respectively. Typically, M is 8 pixels. In the second stage, the biased autocorrelation estimates in Eqn. (4) are used to form the ratios,

$$z_1(k_1,k_2:i,j) = \frac{\hat{R}(k_1,0:i,j)}{\hat{R}(k_2,0:i,j)}$$

$$\qquad (9)$$

$$z_2(\ell_1,\ell_2:i,j) = \frac{\hat{R}(0,\ell_1:i,j)}{\hat{R}(0,\ell_2:i,j)}$$

If we accept the exponential autocovariance model in Eqn. (1), then the expected value of z_1 (say) is [14] ,

$$E\left[z_1\right] = \frac{[\sigma_s^2 \exp\{-k_1\sqrt{\rho_1}\} + (\bar{z}_s)^2][1 - \frac{k_1}{N}]}{[\sigma_s^2 \exp\{-k_2\sqrt{\rho_1}\} + (\bar{z}_s)^2][1 - \frac{k_2}{N}]} \qquad (10)$$

The expected value of z_2 is likewise related to ρ_2. Eqn. (10) represents a nonlinear relationship between z_1 and $\sqrt{\rho_1}$ which can be implemented in the form of a look-up table for the purpose of obtaining ρ_1 values. (See Figure 3.)

Figures 4(a) and 4(b) show the distributions of z_1 and z_2 obtained by applying Eqn. (9) to the autocorrelations of the transformed data $g(i,j)$. The parameters of Eqs. (4), (8) ,(9) and (10) are: $N = 16$, $M = 8$, $\sigma_s^2 = 1500$, $\bar{z}_s = 128$, $k_1 = 1$, and $k_2 = 3$. The histogram distributions of z_1 and z_2 values are depicted in Figure 3 relative to the $z \rightarrow \sqrt{\rho}$ look-up table. Most (80%) of the values fall between the limits 1.154 and 1.194 corresponding to the range $0.0 \leq \sqrt{\rho} \leq 0.25$. This is probably the useful range of ρ for applications in image processing and analysis. Values of z outside this range are caused by breakdown of the exponential model in Eqn. (3). For example, in certain subblocks the autocovariance estimate in Eqn. (5) contains strong periodic components. Interestingly, the mode of the $\sqrt{\rho}$-histogram occurs at $\sqrt{\rho} = 0.1$, corresponding to the correlation coefficient $\exp(-0.1) = 0.9$ often used in nonadaptive processing.

A significant advantage of this new ρ-estimator is that z_1 and z_2 are always positive, thereby avoiding the problem caused by negative autocovariance data in Eqn. (6). Furthermore, the estimates are much less sensitive to very small values of autovariance at lag k_2, caused by random errors in the estimation of autocovariance itself (Eqn. (5)). Also, estimates may be improved by pre-processing the maps of z_1 and z_2 using a smoothing algor-

ithm, thereby reducing the random errors incurred in the model-fitting process. The effect of neighborhood-averaging is evident in the narrowing of the histograms of z_1 and z_2 also shown in Figure 3. (Values of z outside the useable range in Figure 3 may be compressed by means of a slightly modified look-up table. This form of biasing to achieve reduced variance in parameter estimates is common in spectral estimation.) Application of the previous look-up tables (Figure 3) produces the maps of $\sqrt{\rho_1}$ and $\sqrt{\rho_2}$ in Figures 5(a) and 5(b). An important feature of these maps, also apparent in examples later in the paper, is the correlation between the autocovariance parameters and signal activity. It is interesting to note that maps of local variance - often employed for signal measures - are not as sensitive to low contrast detail. The indication is that the nonseparable exponential model for auto-covariance is worthwhile for describing nonstationary data if applied locally.

IV. BEHAVIOR OF THE ESTIMATOR.

This section examines important practical aspects of the behavior of Eqn. (9).

Subblock dimension (N).

Local autocovariance estimation is based on small subblocks of N x N pixels, each one of which is assumed to belong to a global wide-sense station-ary random field. Attaining useful resolution of the image nonstationarity requires that N be small, and yet this violates the normal rules of power spectral estimation. This would suggest, then, that any results of local estimators are useless. However, the fact that our estimates correlate well with observed signal activity suggests the contrary. Further support for Eqn. (9) is now given.

Consider a one-dimensional window of data which contains a single edge (see Figure 6(a)). The biased autocovariance estimate from Eqn. (5) is seen in Figure 6(b) to be piecewise linear. Natural edges tend to be less abrupt, resulting in smoother autocovariance functions which, over small lag values, fit the general shape of the exponential in Eqn. (1). Thus, Eqn. (9) is seen to behave well if N is small enough to traverse single edge-like features. Edge height (contrast) is irrelevant to the estimator. Figure 7 shows maps of $\sqrt{\rho_1}$ and $\sqrt{\rho_2}$ for the well-known "girl" image for the cases N = 16, and 32. The case where N = 16 is near optimum in terms of resolution. (The case N = 8 is excessively noisy.)

Block estimation of parameters.

In certain applications, such as image data compression, single values of the autocovariance parameters are sufficient for representing blocks of N x N pixels, rather than values at every pixel. We may think of the estimation procedure as before with an additional parameter I - the increment between estimates in the (i,j) image space. (I = 1 in all the preceding examples.) The estimation of $\sqrt{\rho}$ values is then posed as follows:

- Apply Eqns. (4), (8), (9) and (10) to give maps consisting of (D/I) x (D/I) values of $\sqrt{\rho_1}$ and $\sqrt{\rho_2}$. (D is the image dimension.)

- Take averages over blocks of dimension (N/I) x (N/I) to give a total of (D/N) x (D/N) values of $\sqrt{\rho_1}$ and $\sqrt{\rho_2}$; i.e., single values of $\sqrt{\rho_1}$ and $\sqrt{\rho_2}$ for each N x N - pixel subblock.

Values of I = 4 and N = 16 have been determined by trial and error to yield results which correlate well with observed signal activity. Greater increments yield insufficient data for the averaging process.

Estimation from degraded images.

The need to estimate autocovariance parameters from images which are blurred and noisy arises in the process of image restoration. Experiments with images blurred by 3 x 3 - neighborhood averaging and additive Gaussian noise (standard deviation 5.0) reveal no appreciable change in the maps of z or $\sqrt{\rho}$. Even noise levels of two or three times this produce useable estimates of the autocovariance parameters.

V. DATA COMPRESSION APPLICATION.

In this section, we show an application of local correlation parameters to block interpolative coding [15]. We address so-called destination interpolation, where the rate of pixels transmitted from image lines is proportional to $\sqrt{\rho_1}$. The procedure is as follows:

- divide the image into N x N - pixel subblocks.

- estimate $\sqrt{\rho_1}$ for each subblock (see previous section).

- subsample each subblock by a function $r = K/\sqrt{\rho_1}$
  to yield a subblock of N x N' pixels. ($K = max [all \sqrt{\rho_1}]$,
  $N' = N/r$).

- transmit the subsampled data, plus the values of $\sqrt{\rho_1}$.

- reconstruct the image by interpolation (the inverse of subsampling.

Subsampling along lines is convenient in images which are raster scanned. Additional compression is possible if across-line redundancy is reduced by

subsampling in this direction using $\sqrt{\rho_2}$ values.

Figure 8 shows the girl image after compression (8(a)) and reconstruction (8(b)) using $\sqrt{\rho_1}$ only. The compressed image consists of contiguous blocks whose width is proportional to $\sqrt{\rho_1}$. Data compression is only 30 % , indicating that this particular data contains little spatial redundancy. The degradation of high frequency details associated with nonadaptive interpolative coding is considerably reduced. Incorporating DPCM coding would reduce the data rate by an additional factor of two.

VI.   CONCLUSIONS.

A simple, useful method for estimating local correlation parameters in images has been presented. The joint assumptions of stationarity within NxN-pixel subblocks (N=16) and the nonseparable exponential model for autocovariance is seen to work well when judged on the basis of comparisons with observed signal activity. The resulting maps of autocovariance parameters are useful in image data compression and, we expect, in adaptive filtering and restoration.

ACKNOWLEDGMENTS

REFERENCES

[1] C. W. Helstrom, "Image Restoration by the Method of Least-Squares,"
    J. Opt. Soc. Am., Vol. 57, pp. 297-303, 1967.

[2] J. B. O'Neal, "Predictive Quantizing Systems (Differential Pulse Code
    Modulation) for the Transmission of Television Signals," Bell
    System Tech. Journal, Vol. 45, May-June, 1966.

[3] M. J. Lahart, "Local image restoration by a least-squares method,"
    J. Opt. Soc. Am., Vol. 69, No. 10, pp. 1333-1339, October, 1979.

[4] G. L. Anderson and A. N. Netravali, "Image Restoration based on a Sub-
    jective Criterion," IEEE Trans. Syst, Man, and Cyber, Vol. 6,
    pp. 845-853, 1976.

[5] R. Kasturi, "Adaptive Image Restoration in Signal-Dependent Noise,"
    Ph.D. dissertation, Inst. for Electronic Science, Texas Tech Uni-
    versity, Lubbock, Texas, 1982.

[6] G. K. Froehlich, J. F. Walkup, R. B. Asher, "Optimal estimation in
    signal-dependent noise," J. Opt. Soc. Am., Vol. 68, No. 12, pp. 1665-
    1672, December, 1978.

[7] A. Habibi, "Survey of Adaptive Image Coding Techniques," IEEE Trans.
    Comm., Vol. 25, pp. 1315-1322, 1977.

[8] B. Widrow, "Adaptive Filters I: Fundamentals," Stanford Elec. Lab.
    Report, SEL-66-126, December, 1966.

[9] A. K. Jain, "Advances in Mathematical Models for Image Processing,"
    Proc. IEEE, Vol. 69, No. 5, pp. 502-528, May, 1981.

[10] G. E. Box, and G. M. Jenkins, Time Series Analysis Forecasting and
     Control, Holden-Day, San Francisco, California, 1971.

[11] R. B. Blackman and J. W. Tukey, The Mesaurement of Power Spectra, Dover,
     New York, 1959.

[12] R. N. Strickland, "Transforming Images into Block Stationary Behavior,"
     Applied Optics, Vol. 22, pp. 1462-1473, May, 1983.

[13] R. Wallis, "An approach to the Space Variant Restoration and Enhancement
     of Images," in Proceedings, Symposium on Current Mathematical Prob-
     lems in Image Science, Naval Postgraduate School, Monterey, California,
     November, 1976.

[14] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, (Prentice-
     Hall, Englewood Cliffs, N. J., 1975), pp. 540.

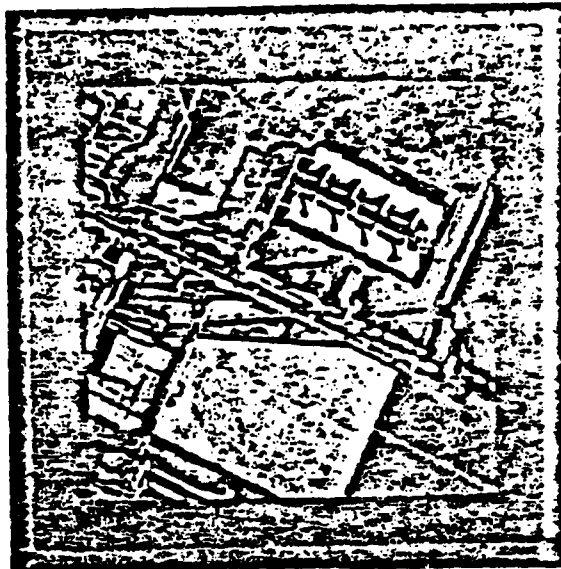[15] W.K.Pratt, Digital Image Processing , (Wiley,New York,1978), pp.665.
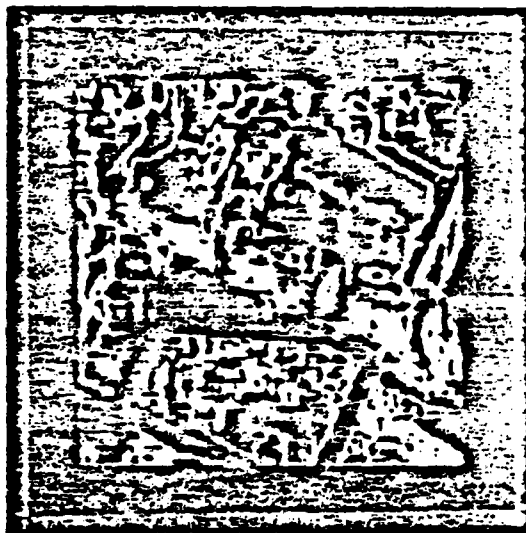
Figure 1. Test image "Industry"
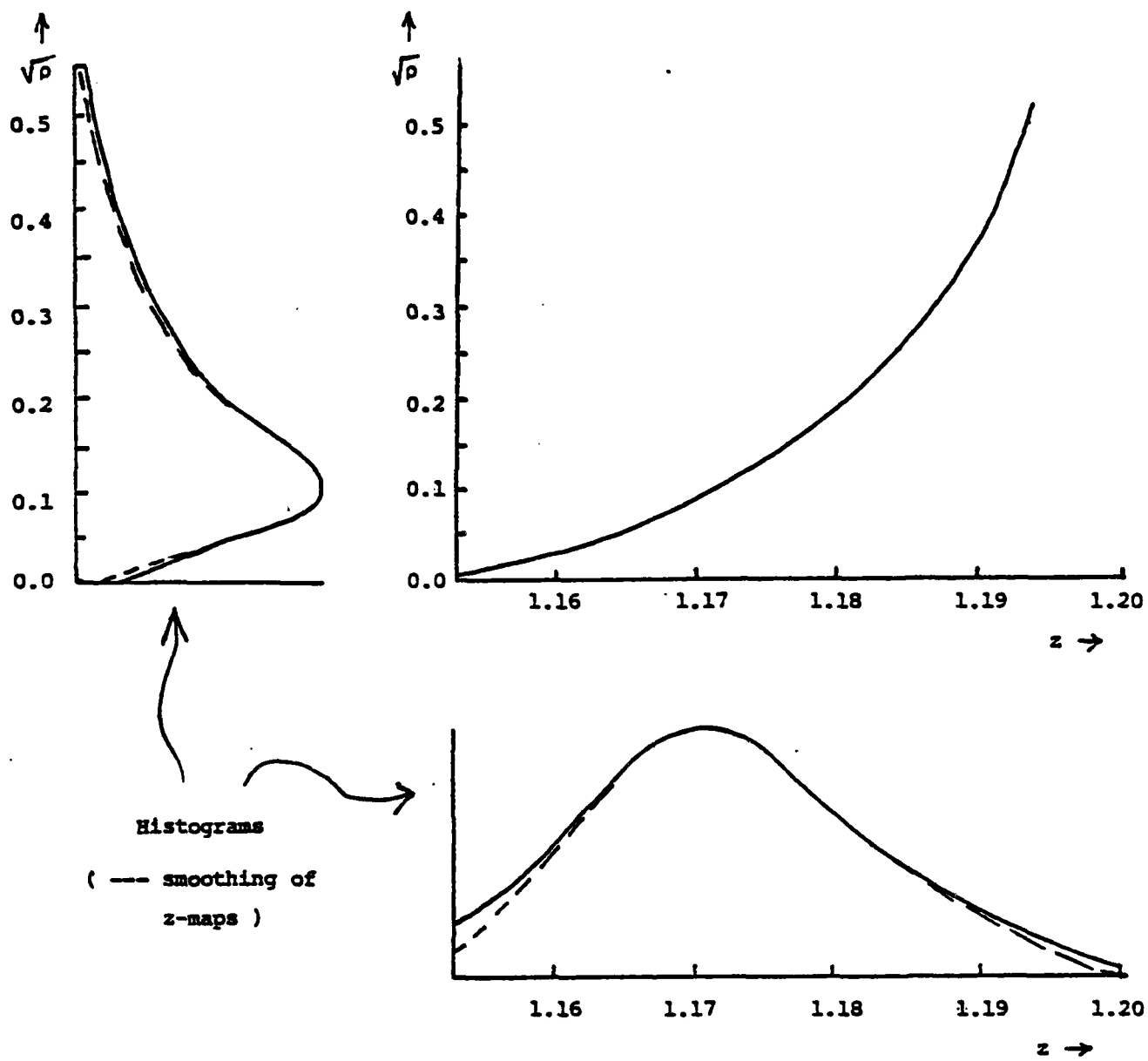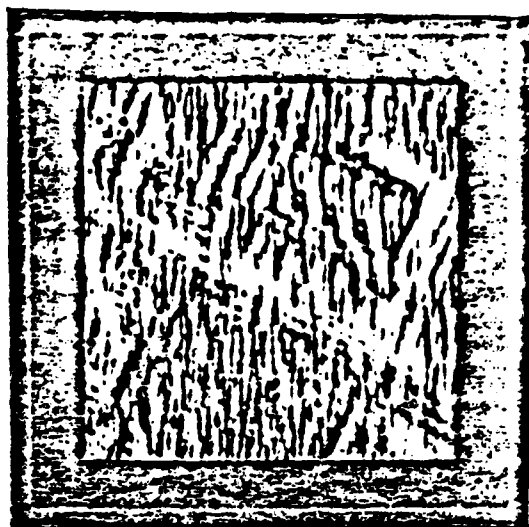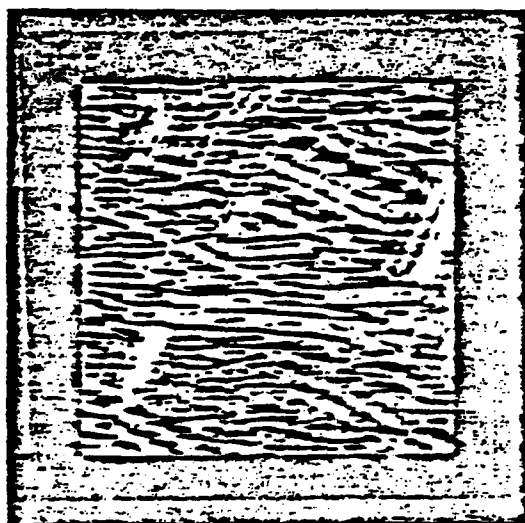

Figure 2. Map of $\sqrt{\rho_1}$ using Eqn.(6).

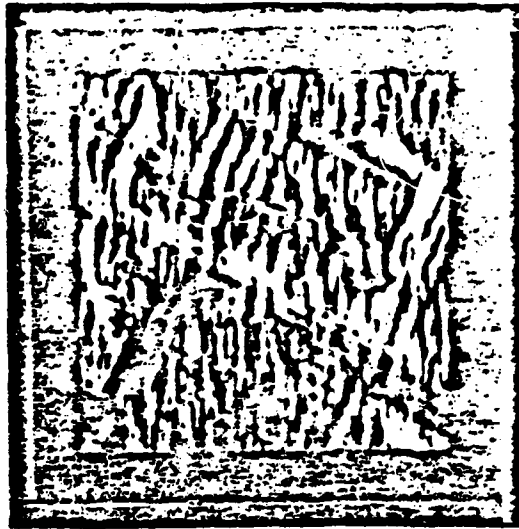Figure 3. Look-up table for z $\rightarrow \sqrt{\rho}$ mapping.
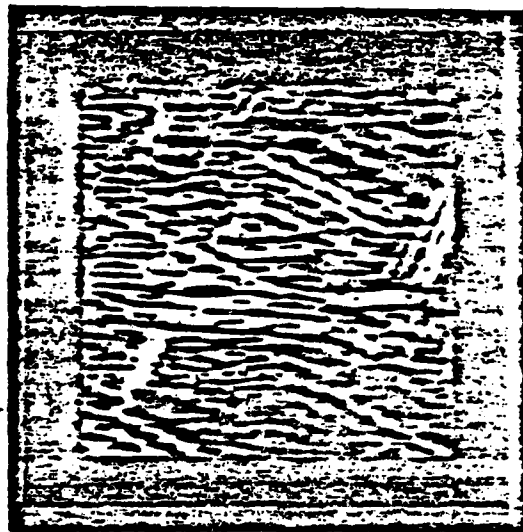
(a) Map of z_1 from Eqn.(9).

(b) Map of z_2 from Eqn.(9).

Figure 4. Maps generated by applying Eqn.(9) to "Industry" image.

(N=16, M=8, σ_s^2=1500, $\bar{\bar{f}}_s$=128, $k_1=\ell_1$=1, $k_2=\ell_2$=3)

(a) Map of $\sqrt{\rho_1}$ derived from z_1 in Figure 4(a).
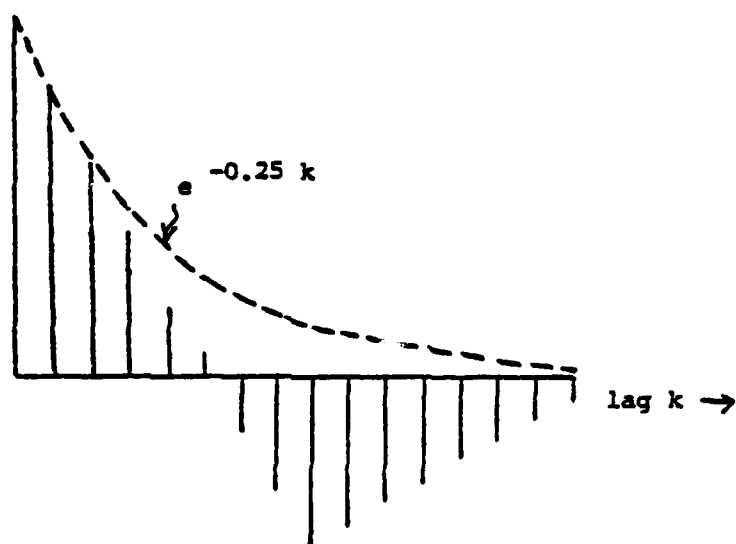

(b) Map of $\sqrt{\rho_2}$ derived from z_2 in Figure 4(b).

Figure 5. Autocovariance parameter maps from "Industry" image.

(SCALE: (black) $0.0 \leq \sqrt{\rho} \leq 0.5$ (white)).

(a) 16 samples across an edge.
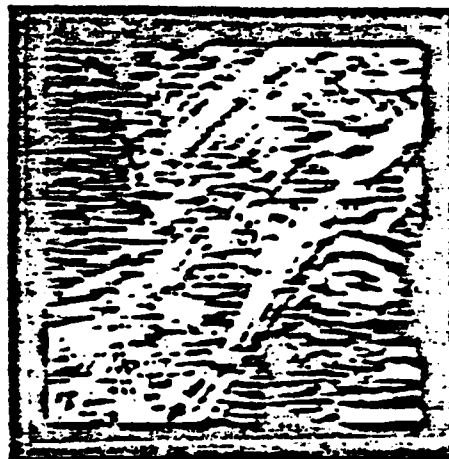
$$e^{-0.25\ k}$$

lag k →

(b) Autocovariance of (a).

Figure 6. Comparison of autocovariance of a single edge with the exponential model, showing that lags 0 – 3 are a good fit.

(a)"Girl" image.

(b) Map of $\sqrt{\rho_1}$, N = 16.

(c) Map of $\sqrt{\rho_2}$, N = 16.

(d) Map of $\sqrt{\rho_1}$, N = 32.

(e) Map of $\sqrt{\rho_2}$, N =32.

Figure 7. Autocovariance parameter maps with

N varying.

(a) Compressed

(b) Reconstructed

Figure 8. Application to block-interpolative destination coding.

(VI.) LIST OF PUBLICATIONS RESULTING FROM AFOSR SUPPORT

(1) B. R. Hunt, "Some remaining problems in nonlinear image restoration", Proc. Symp. Image Science Mathematics, Navel Postgrad. School, Monterey, Calif. Nov. 10-12, 1976.

(2) B. R. Hunt, "An optical analogy to DPCM digital image data compression", Proc. SPIE, Vol. 118, August, 1977.

(3) D. G. McCaughey, "An image coding algorithm using spline functions", Proc. SPIE, Vol. 149, August, 1978.

(4) D. G. McCaughey and H. C. Andrews, "Image app.oximation by variable knot bicubic splines".

(5) B. R. Hunt, "Optical computations for image bandwidth compression: analysis and simulation", Appl. Optics, Vol. 17, pp. 2944-2951, 1978.

(6) D. J. Granrath and B. R. Hunt, "Signal detection trade-off analysis of optical vs. digital Fourier transform computers", Appl. Optics, Vol. 18, pp. 36-43, 1979.

(7) D. J. Granrath and B. R. Hunt, "A two-channel model of image processing in the human retina", Proc. SPIE, Vol. 199, August, 1979.

(8) B. R. Hunt, "Nonstationary statistical image models", Computer Graphics and Image Processing, Vol. 12, pp. 173-186, 1980.

(9) D. G. McCaughey, "Variable resolution hybrid-coding techniques", ASSP Workshop on Two-Dimensional Signal Processing, Berkeley, 1979.

(10) D. G. McCaughey and H. C. Andrews, The continuous-discrete model: least-squares inverses and singular function expansions".

(11) B. R. Hunt and S. D. Cabrera, "Optical implementation of a
      spatially adaptive image data compression scheme", _Proc._
      SPIE, Vol. 232, Washington, 1980.

(12) B. R. Hunt, and H. Ito, "A hybrid optical/digital inter-
      frame image data compression scheme", _Proc. SPIE_, Vol. 249,
      San Diego, 1980.

(13) B. R. Hunt and S. D. Cabrera, "Optical computations for a
      spatially adaptive image data compression system", _Optical_
      Engineering, Vol. 20, pp. 616-620, 1981.

(14) H. Ito and B. R. Hunt, "A temporally adaptive hybrid optical/
      digital interframe compression scheme", _Proc. SPIE_, Vol. 292,
      San Diego, 1981.

(15) W. R. Stevens and B. R. Hunt, "Software pipelines in image
      processing", _Computer Graphics and Image Processing_, Vol.
      20, 1982.

(16) R. N. Strickland, "Adaptive data compression by transforma-
      tions for generating stationary statistical image models",
      Proceedings of the York Conference on Image Processing, York,
      U. K., 1982.

(17) R. N. Strickland, "Transforming images into statistically
      stationary behavior", accepted for publication, _Applied Optics_.

(18) W. Smith and R. N. Strickland, "Adaptive data compression by
      statistically stationary transformations", accepted for
      publication, _Applied Optics_.

(19) B. R. Hunt, O. Kubler, "An optimal theory of multispectral
      image restoration", _SPIE Proc._, Vol. 397, Geneva, April 1983.

END

FILMED

DTIC
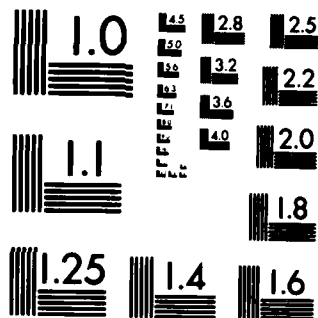
MICROCOPY RESOLUTION TEST CHART

(20) S. Park, R. Schowengerdt, "Image reconstruction by parametric cubic convolution", Computer Graphics and Image Processing, Vol. 20, 1982.

(21) R. Schowengerdt, S. Park, R. Gray, "Topics in the 2-dimensional sampling and reconstruction of images", Int'l Journal of Remote Sensing, March 1983.

(22) S. Park, R. Schowengerdt, "Image sampling, reconstruction, and the effects of sample-scene phasing", Applied Optics, Vol. 21, Sept. 1982.

(23) D. Fraser, B. Hunt, J. Su, "Principles of tomography in image data compression," accepted for publication in Optical Engineering.

(24) Z. Cao, B. Hunt, "Improvements to IDPCM and its combination with transform coding," under review for publication, Optical Engineering.

(25) Z. Cao, B. Hunt, "Image data compression by recursive IDPCM," under review for publication, Optical Engineering.

(VII.) Personnel

The following is a list of personnel who received partial support from Grant AFOSR-81-0170 during the period from March 15, 1983 to March 14, 1984.

Professor B. R. Hunt: Principal Investigator.

Assistant Professor R. N. Strickland: Department of Electrical and Computer Engineering.

Assistant Professor R. Schowengerdt: Remote Sensing and Department of Electrical and Computer Engineering.

Dr. Robert Gray: completed Ph.D degree in September, 1983.

Mr. Keith Morgan: Ph.D candidate in Electrical and Computer Engineering.

Ms. Karen West: Ph.D candidate in Applied Mathematics.

Mr. Fu Deng-Yuan: M Sc. candidate in Electrical and Computer Engineering.

Ms. Marcia Collaer: M Sc. candidate in Electrical and Computer Engineering.

END

FILMED

4-85

DTIC